

Supporting Bio-Medical Knowledge Discovery: the Archetype-Based Electronic Bio-Medical Research Record (eBMRR)

Thesis

submitted to the
University of Dublin, Trinity College
in fulfillment of the requirements
for the degree of
Doctor of Philosophy

Melanie Bettina Späth, M.Sc., B.A. (Mod.)
Centre for Health Informatics,
Department of Computer Science,
Trinity College,
Dublin, Ireland.

March 2013

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is entirely my own work.

Melanie Bettina Späth

March 4, 2013

Summary

This thesis proposes the development of a novel type of record, the electronic Bio-Medical Research Record, which generically integrates molecular research and patient data in an interoperable manner to support bio-medical knowledge discovery in biobanks.

To support bio-medical knowledge discovery, clinical patient information, such as medical history and lifestyle information, needs to be combined with results from molecular research, such as omic information won from novel high-throughput analysis techniques on patient samples. While the former can be found in the patients' medical records, the latter may be scattered over researchers' computers in a variety of proprietary data formats, hampering the integration of information within a research setting and the sharing of this information between collaborating researchers and its preservation for future integrative research.

Biobanks or bio-repositories play an important role at the intersection of these two streams of information. Biobanks are places that collect, store and distribute biological specimens, such as blood, urine and tissue and relevant patient data, such as clinical history and life-style information. This information is stored in the biobank information management system (BIMS). However, current BIMS solutions tend to be proprietary, disease and study-specific, so that major resources need to be invested in setting up a BIMS for a new study or biobank and information is not easily shareable or available for reuse in future research.

Therefore, the requirements for the eBMRR are that it be easily adaptable to any biobank or study, thus easing the process of setting up a new biobank information management system and that it supports the sharing and integration of different types of clinical and research information within and across biobanks to support the sharing of research data and large-scale integrative data analysis.

In this research, the feasibility of using the *openEHR* approach for the realisation of this record is explored. The *openEHR* approach is a recent standards-based approach towards the development of interoperable and future-proof EHRs, defining a novel approach to generically model clinical information in reusable archetypes to support the development of syntactic and semantic interoperable EHRs.

The tasks involved in this research are two-fold. First, clinical archetypes developed for use in a healthcare environment are tested for their reusability in the research context to record a patient's background information in a biobank setting. To do so, the database schema of the biobank of the Irish Prostate Cancer Research Consortium

(PCRC) is taken as model and a mapping and matching of the database model in the PCRC biobank to existing archetypes is carried out, using an *openEHR* data modelling approach. The outcome of this part is that clinical archetypes can be reused in a research context to support the recording of a patient’s medical background information, thus reducing modelling effort while ensuring both syntactic and semantic interoperability, allowing for the safe sharing of clinical background information between researchers. This research also gives an insight into the challenges involved with regard to integrating legacy data in biobanks into an *openEHR* based system. These challenges are more complex due to integrating data recorded with a different purpose (research use vs. supporting direct healthcare), but most are commonly encountered legacy integration challenges.

Secondly, the development of archetypes to model omic data is investigated. A semi-automatic approach to leverage recent omic data standard developments is proposed and applied. The creation of these archetypes included the development of bespoke transformation software followed by a semantic review process, complying with *openEHR* editorial guidelines. The outcome of this approach is the successful modelling of sets of generic omic archetypes that support the recording of the execution and result of omic experiments on patient samples, which has been carried out for mass spectrometry and gel electrophoresis experiments. The syntactic correctness of these newly developed archetypes was evaluated with available archetype tools, in addition, their semantic correctness was evaluated by arranging the archetypes in templates and mapping them to omic instance data provided in the original data formats. Thus, the eBMRR can be composed of clinical and research archetypes, allowing researchers to share both clinical background and omic research information on patient samples in a syntactically and semantically interoperable manner, while reducing modelling effort in biobanks. The finding of this research is that the *openEHR* archetype approach can be used to model omic information. However, a number of limitations upon using the *openEHR* archetype approach to model omic data and of using a patient-centric *openEHR* based record in a research environment are identified during this process that need to be addressed in future research.

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Melanie Bettina Späth

March 4, 2013

Acknowledgements

There are several people without whom this work would not have been possible.

First and foremost, I would like to thank my supervisor, Prof. Jane Grimson for her support during my research. I will always be indebted to her for her encouragement, constructive criticism and unwavering belief in me and my work. She has been supportive beyond her duties and I am forever grateful to her for that. Without her positive attitude and never-ending encouragement I would not have been able to finish my work. I am also very thankful to her for allowing me to follow my passion for cycling in parallel - without her blessing I would not have been so successful in it.

I would also like to thank a few other people that have helped me during my PhD. For expert advice and engagement in valuable discussions on the archetype approach I would like to thank Damon Berry. In his words - “there are not many people who are connected through an odd passion for archetypes”. Similarly, I would like to thank my friend Alan Donnelly for lending me an open ear for my ideas, many late night discussions and mental support when the going got tough. You are a real friend.

Finally, I want to thank my husband, Ryan Sherlock. Ryan, without your love, kindness, encouragement and support I would never have been able to do this. For several years now he has patiently endured and listened through my problems. He’s always been there when I needed him and has kept me motivated when I was doubting myself. There is no more that I could ask of him.

MELANIE BETTINA SPÄTH

*University of Dublin, Trinity College
March 2013*

Contents

Summary	iv
Acknowledgements	ix
List of Tables	xvi
List of Figures	xvii
List of Publications & Presentations	xviii
Chapter 1 Introduction	1
1.1 Motivation and Justification	1
1.2 Research Question	4
1.3 Objective, Goals and Scope of the Thesis	5
1.3.1 Objective 1: Reusability of clinical archetypes	5
1.3.2 Objective 2: Development of omic archetypes	5
1.4 Research Approach	6
1.5 Contribution to State of the Art	6
1.6 Thesis Overview	7
Chapter 2 State of the Art	9
2.1 Introduction	9
2.2 Standardization and integration efforts for the sharing of clinical data .	9
2.2.1 Emergence of standards for the sharing of clinical data	9
2.2.2 ISO 18308	10
2.2.3 CONTSYS (EN 13940)	11
2.2.4 HISA (EN 12967)	11
2.2.5 GPIC (EN 14822)	12
2.2.6 ASTM healthcare standards	12
2.2.7 CDISC	13
2.2.8 IHE	14
2.2.9 HL7	15
2.2.10 EN 13606	16
2.2.11 The two-level modelling methodology	16
2.2.12 <i>openEHR</i>	22

2.2.13	Relationship between EN 13606, <i>openEHR</i> and HL7v3	29
2.2.14	Controlled vocabularies, terminologies and ontologies in the clinical domain	31
2.2.15	Summary of EHR standard developments	32
2.3	Biobanking	33
2.3.1	Introduction	33
2.3.2	The Biobank Information Management System	34
2.3.3	Collaboration between biobanks	35
2.3.4	Summary of Biobanking	37
2.4	Sharing omic data	38
2.4.1	Introduction	38
2.4.2	Short history of omic data reporting standards	39
2.4.3	Reporting standards in Transcriptomics	40
2.4.4	Reporting standards in Proteomics	46
2.4.5	Reporting standards in Genomics	52
2.4.6	Standard development in Metabolomics	54
2.4.7	Collaboration, coordination and harmonisation in the development of omic standards	55
2.4.8	Summary of omic data sharing	65
2.5	Summary	67

Chapter 3 Modelling the database of a biobank information management system with archetypes 69

3.1	Introduction	69
3.2	The PCRC biobank	70
3.3	The meta-data of the PCRC biobank information management system	72
3.4	Mapping of data types	77
3.5	Semantic mapping	78
3.6	Archetypes modelling the data in the PCRC biobank	84
3.7	Design of templates for the PCRC BIMS	89
3.8	Summary	95

Chapter 4 Development of ‘omic’ archetypes 97

4.1	Introduction	97
4.2	Leveraging omic data standards in the development of omic archetypes	98
4.3	Creating archetypes from omic data exchange formats	105
4.3.1	Phase 1: Syntactic transformation	105
4.3.2	Phase 2: Semantic review	107
4.4	Development of omic archetypes for mass spectrometry	110

4.4.1	The mzML data exchange format	110
4.4.2	Mapping of mzML structures to archetype structures	116
4.4.3	Automation of the syntactic transformation process	127
4.4.4	Syntactic transformation of mzML	128
4.4.5	Semantic review of candidate archetypes for mass spectrometry	128
4.5	Development of omic archetypes for other omic data formats	149
4.5.1	Introduction	149
4.5.2	The FuGE data exchange format	150
4.5.3	Syntactic transformation of FuGE	153
4.5.4	Semantic review of candidate archetypes for functional genomics	157
4.5.5	Summary of the creation of archetypes from FuGE	169
4.5.6	The GelML data exchange format	170
4.5.7	Syntactic transformation of GelML	172
4.5.8	Semantic review of candidate archetypes for gel electrophoresis	172
4.5.9	Summary of the creation of archetypes from GelML	178
4.6	Summary	178
Chapter 5 Evaluation and Discussion		181
5.1	Introduction	181
5.2	Evaluation of modelling the information in the PCRC database with <i>openEHR</i> archetypes and templates	182
5.2.1	Modelling challenges	182
5.2.2	Difficulties identifying matching candidate archetypes	187
5.2.3	Using clinical archetypes to describe clinical information in a research context	189
5.2.4	Issues mapping legacy database to archetypes	192
5.2.5	Summary of mapping issues	198
5.2.6	Further observations regarding new archetypes created in this research	200
5.3	Summary of modelling the information in the PCRC database with <i>openEHR</i> archetypes and templates	200
5.4	Evaluation of modelling omic information with the <i>openEHR</i> archetype methodology	203
5.4.1	Syntactic validity of the developed omic archetypes	204
5.4.2	Semantic validity of the developed omic archetypes	206
5.4.3	Evaluation of mass spectrometry archetypes	208
5.4.4	Evaluation of gel electrophoresis archetypes	215
5.4.5	Evaluation of FuGE-based archetypes	221

5.4.6	Evaluation of the process used to create omic archetypes	222
5.4.7	Limitations of using the <i>openEHR</i> archetype approach to model omic information	228
5.5	Summary of modelling omic information with <i>openEHR</i> archetypes and templates	240
5.6	Summary	242
Chapter 6	Conclusions and Future Work	245
6.1	Introduction	245
6.2	Objectives and Achievements	246
6.2.1	Objective 1: Reusability of clinical archetypes in a research context	247
6.2.2	Objective 2: Development of omic archetypes	248
6.2.3	Reflection	251
6.2.4	Conclusion	251
6.3	Contributions to our knowledge	252
6.3.1	Reusability of clinical archetypes	252
6.3.2	Approach for modelling data with archetypes	253
6.3.3	Integration of legacy systems with archetypes	253
6.3.4	Archetypes and templates for the PCRC biobank	254
6.3.5	Strategy for developing omic archetypes	254
6.3.6	Development of generic omic archetypes	254
6.3.7	Using the <i>openEHR</i> archetype approach to model non-clinical information	255
6.4	Future Work	255
6.4.1	Consultation with research scientists	255
6.4.2	Creation of archetypes from further omic data formats	255
6.4.3	Investigation on extending the reference model	256
6.4.4	High-level organisation of the eBMRR	256
6.4.5	Integration of ontologies and CVs	257
6.4.6	Development of device plug-ins	258
6.5	Summary	258
References		260
Appendices		284
Appendix A	Details of the PCRC BIMS database schema	285
Appendix B	The MIAPE-MS guidelines and checklist	293

Appendix C	Automation of the syntactic transformation process	297
Appendix D	Archetype checkpoints	302
Appendix E	List of candidate archetypes from mzML	303
Appendix F	List of candidate archetypes from FuGE	306
Appendix G	List of candidate archetypes from GelML	310
Appendix H	<i>openEHR</i> system implementations	315

List of Tables

3.1	Mapping of the BIMS data types to <i>openEHR</i> data types	77
3.2	Modifications carried out on existing archetypes	87
3.3	Summary of archetypes used to cover the PCRC BIMS concepts	88
3.4	Summary of templates for the PCRC BIMS	90
4.1	Mapping of XML data types to <i>openEHR</i> data types	119
4.2	Data types specified in PSI-MS ontology	120
4.3	CVMappingRules and object occurrences	126
4.4	CVTerm attributes and object occurrences	126
4.5	Comparison of overlapping archetypes for specimen	139
4.6	Component archetypes	140
4.7	Comparison of CLUSTER.source_file vs. DV_MULTIMEDIA	141
4.8	Comparison of CLUSTER.binary_data_array vs. DV_MULTIMEDIA	142
4.9	List of final mass spectrometry archetypes	148
4.10	Existing archetypes needed for full mass spectrometry template	149
4.11	List of remaining FuGE-based candidate archetypes	161
4.12	List of final functional genomics archetypes	169
4.13	List of final gel electrophoresis archetypes	178
A.1	Explanation of the fields in the tables of the PCRC BIMS database	292
D.1	Archetype checkpoints	302
E.1	List of mzML candidate archetypes	305
F.1	List of FuGE candidate archetypes	309
G.1	List of GelML candidate archetypes	314
H.1	Comparison of Implementations	317

List of Figures

2.1	Scope of the HISA standard	12
2.2	Single-level information system development methodology	17
2.3	Two-level information system development methodology	17
2.4	Hierarchy of the ISO/EN 13606 Reference Model	18
2.5	Diagram of the EHR Extract hierarchy	19
2.6	Blood pressure archetype	20
2.7	Screenshot of archetype generated screen form	21
2.8	Relationship between archetypes and templates	21
2.9	Single and Multiple-valued C_ATTRIBUTES	24
2.10	ADL Archetype Structure	26
2.11	Schematic relationship between <i>openEHR</i> , EN 13606, and HL7 CDA .	30
2.12	Schematic representation of the six components of a microarray experi- ment.	42
2.13	Schematic representation of the six components of MISFISHIE.	46
2.14	Schematic overview about the MlXS standard	53
2.15	Hierarchical organisation of MIGen	54
2.16	Interaction graph for MIBBI projects	57
2.17	RSBI top-level concepts and their relationships	59
2.18	The package structure of FuGE	62
2.19	Representation of biomedical investigation using OBI Core	65
3.1	Overview of the hierarchical PCRC BIMS database table relationships .	73
3.2	Part 1 of the PCRC BIMS database	74
3.3	Part 2 of the PCRC BIMS database	74
3.4	Part 3 of the PCRC BIMS database	75
3.5	Flow-chart of modelling the data in the BIMS database with archetypes	78
3.6	Re-grouping of BIMS concepts (leaf nodes collapsed)	81
3.7	Re-grouped BIMS fields	82
3.8	Screenshot of “Sample Collection - Urine.oet” template	91
3.9	Screenshot of “Sample Collection - Urine.oet” screen form	92
4.1	2-phase process for creating omic archetypes	109
4.2	Mass spectrometer	111

4.3	Mass spectrum	111
4.4	Structure of mzML	113
4.5	Format of the CvMappingRule file	114
4.6	Relationship between ITEM, CLUSTER and ELEMENT	118
4.7	The hierarchy of base classes in FuGE	151
4.8	The package structure of FuGE	151
4.9	The FuGE containment hierarchy	152
4.10	FuGE classes for representing experimental procedures	163
4.11	FuGE classes for representing a protocol	163
4.12	FuGE class for representing the running of a protocol.	164
4.13	Relationships between Protocols, Actions, ProtocolApplications and ActionApplications	165
4.14	Creating archetypes for FuGE classes: Absorbing Action and ActionApplication into parent classes	166
4.15	Creating archetypes for FuGE classes: Creating slots for ChildProtocols and ChildProtocolApplications	167
4.16	Creating archetypes for FuGE classes: Combining Protocols with ProtocolApplications	168
4.17	Product of a 2-D electrophoresis experiment	170
4.18	A graphical representation of GelML	171
5.1	Screenshot of a template for a spectrum example	212
5.2	Screenshot of a template for a chromatogram example	215
5.3	Screenshot of a template for one-dimensional electrophoresis	218
5.4	Screenshot of a template for two-dimensional electrophoresis	221
5.5	Screenshot of a FuGE template	222

List of Publications

- **Melanie Bettina Späth, Jane Grimson.** Applying the archetype approach to the database of a biobank information management system. *International Journal of Medical Informatics*, 80(3): 205-226 (2011)
- **Melanie Bettina Späth, Jane Grimson.** Supporting biomedical knowledge discovery: The electronic Biomedical Research Record (eBMRR). Presented at the *14th Annual Conference of the Health Informatics Society of Ireland (HISI)*, 18th and 19th November 2009. The paper received the “Best Paper Award” at the conference.
- **Melanie Bettina Späth, Jane Grimson.** Development of omic archetypes for the electronic Biomedical Research Record. Presented at the *16th Annual Conference of the Health Informatics Society of Ireland (HISI)*, 16th and 17th November 2011.

Chapter 1

Introduction

1.1 Motivation and Justification

The primary purpose of electronic health records (EHRs) is to provide a documented record of care to support present and future healthcare [1]. However, EHRs can also serve as an important source of information for scientific research and for candidate selection for clinical trials [2]. Bio-medical knowledge discovery, such as the investigation of the fundamental disease mechanisms of complex diseases, requires researchers to combine clinical patient data, such as medical history and lifestyle data found in the patients' medical records with the results from molecular experiments on human biospecimens [3]. Biobanks, or bio-repositories play a central role in combining these two streams of information [4]. Biobanks are places that collect, store and distribute biological specimens, such as blood, urine and tissue and associated patient data, such as clinical history and life-style information. Recent advances in biotechnology, especially the development of high-throughput technologies, such as microarrays, comparative genomic hybridisation, 2D gel electrophoresis, mass spectrometry identification, metabolic profiling and other techniques that produce large amounts of research data are revolutionising biomedical research with regard to the speed and scale at which it can be carried out. As a consequence of increased research activity due to these advances, the demand for high-quality, well-annotated human biospecimens in biomedical research [5, 6, 7, 8, 9, 3] has increased. To fulfil this need, new biobanks are being set up all over the world and biobanking activity is generally increasing [10]. Characteristically, the establishment of new collections is often motivated by specific diseases and small sample collections usually outnumber larger ones [10].

An accurate collection of detailed phenotypic/clinical patient information, such as medical history of the sample donor, is necessary so that correlations can be drawn about the influence of genes and/or the environment on disease pathology [11, 12, 13, 14]. Furthermore, the application of high-throughput technologies, has led to an almost exponential increase in the information created in bio-medical research [14], such as genomic, proteomic and other “-omic” data. The management of this data

in biobanks and its integration with existing health records present considerable challenges [14]. The data in biobanks is managed by the biobank information management system (BIMS). The BIMS stores information about the samples, such as the composition of the sample and sample handling and administrative information which is linked to the clinical patient information, such as the disease, treatment and patient outcome [15]. Current BIMS solutions and their underlying databases tend to be bespoke disease and study specific proprietary implementations of variable sophistication (e.g. [13, 14]), reflecting the heterogeneity of biobanks. Major resources and effort are invested in the development of bespoke BIMS solutions whenever a new study is carried out or a new biobank is being established. Furthermore, due to the heterogeneity of solutions data cannot easily be shared or integrated with larger datasets between collaborating biobanks to achieve statistically significant results. Research is carried out in an isolated manner and the raw data from sample analyses on patient samples is scattered in a variety of incompatible and/or proprietary data formats over researchers' computers hindering integrative analysis, sharing and reuse of this data for further research. This severely restricts the scale and amount of research that can be carried out. For example, for translational cancer research, large numbers of cancer patients, together with high-quality biosamples from these patients and associated patient data are needed to be able to achieve statistically significant results in a sensible time frame [3]. However, differences in the collection, annotation and storage of data about the patient and their samples and experimental results in different biobank settings prevent researchers from being able to easily combine information in studies and share research results that may have been expensive and time-consuming to produce. Samples and patient background information may be recorded in different, possibly incompatible ways with regard to granularity, terminology used and organization of the data. Attempting to combine data from heterogeneous sources is at best work intensive and time-consuming, as shown during the creation of a federated twin database, TwinNET [16] and at worst impossible, severely restricting the large-scale analyses over disparate sets of data needed for translational research.

This research proposes the development of a novel type of record, the electronic Bio-Medical Research Record (eBMRR) for biobanks, which generically integrates molecular research and patient data in an interoperable and future-proof way to support support the investigation of fundamental disease mechanisms and treatment processes in biobanks and ease the development effort whenever a new biobank or study is set up. In order to integrate such diverse data in a single record system and make it safely shareable, it is necessary that the information to be recorded is modelled consistently. In addition, the information needs to be modelled in a way that it is widely applicable so that the record can be customised to different settings. Modelling information in

large, complex and evolving domains is difficult [17]. To achieve interoperability, information modelling needs to balance specificity and genericity, so that the same model can be used in different contexts to record information while being specific enough to allow for safe sharing of that information.

Attempts to solve the problem of integrating and sharing different types of information safely and reliably have been researched for many years in the clinical domain to support health professionals in patient care and more recently in the omic domain to support collaboration and reuse of research data that was expensive to produce in future large-scale integrative research.

In the context of electronic health/patient records, past attempts were largely proprietary focusing on particular diseases e.g. creating an epilepsy or a diabetic patient record. However, newer approaches focus on the development of standards-based solutions to facilitate sharing of clinical data between healthcare organisations more easily and efficiently, thereby reducing development effort. One such approach is the archetype approach developed by *openEHR*[17]/CEN, based on the CEN/ISO 13606 standard. The *openEHR* archetype approach aims to support the development of semantically interoperable and future-proof EHRs [17]. In this approach information and knowledge are modelled on two different levels [17]. The first level models the stable semantics of information in a small reference model. The second level models specific domain knowledge concepts, for example clinical concepts such as “progress note” or “blood pressure” in reusable archetypes, by constraining the information building blocks in the reference model. A consequence of this approach is that archetype-based EHRs are interoperable regardless of the software used to implement them, since the reference and archetype models defined in this approach are software and platform independent [17]. Another consequence of this approach is that it enables domain experts to easily model domain knowledge concepts in archetypes according to the information recording requirements of their domain, without needing to have knowledge of underlying technologies used in the implementation [17]. Furthermore, archetypes are generic and reusable and can easily be adapted to the information requirements at hand and a number of clinical archetypes modelling clinical information already exist.

Similarly, in the domain of omic research, past attempts at recording information have resulted in a number of proprietary data formats. However, there have been many recent community-led initiatives at defining domain-specific data reporting standards and associated data formats, such as the “Minimum Information About a Microarray Experiment” (MIAME) specification [18] and associated MicroArray Gene Expression Markup Language (MAGE-ML) [19] defined by the Microarray Gene Expression Data Society (MGED) (now the Functional Genomics Data Society (FGED))¹

¹MGED/FGED, www.mged.org/

in transcriptomics, the “Minimum Information About a Proteomics Experiment” (MI-APE) specification [20] and associated proteomics data formats (for example mzML for mass spectrometry [21]) developed by the Human Proteome Organisation (HUPO) Proteomics Standards Initiative (PSI)² in proteomics and the Minimum Information about any (x) Sequence (MIxS) standard [22] and the associated Genomic Contextual Data Markup Language (GCDML) [23] developed by the Genomic Standards Consortium (GSC)³ in genomics to support the safe sharing and reuse of omic research data between researchers.

This thesis explores the feasibility of basing the development of the proposed eBMRR on the *openEHR* archetype approach, leveraging the work that has already been carried out in the development of clinical archetypes and applying the approach to model omic data, leveraging the recent emergence of omic reporting standards in the research community for the development of omic archetypes. Together with advances in information technology, translational research from bench to bedside is becoming feasible. These advances promise to provide the means to enable truly personalised, life-long healthcare with targeted and individualised treatments and health promotion plans, aiming to reduce disease, promote health and improve quality of life [4, 13]. Being able to share and integrate clinical patient and research information in a generic way as proposed in the development of the eBMRR is central to realising this vision.

1.2 Research Question

This research explores the feasibility of using the *openEHR* archetype methodology [17] to develop the proposed electronic Bio-Medical Research Record. Thus, the research question is:

“Is it feasible and beneficial to use the openEHR approach for the development of the electronic biomedical research record that integrates clinical and research information in a generic way?”

The underlying assumption is that such a record will facilitate the sharing of integrated clinical (phenotypical) and research (such as genomic) data between biomedical researchers within and between biobanks in the same way that the archetype-based EHR supports the safe sharing of healthcare information between clinicians. Just as the EHR improves healthcare delivery, the proposed integrated research record seeks to enhance knowledge discovery by supporting the integration of different types of data from multiple biobanks, thus allowing large-scale integrative research.

²HUPO PSI, www.psdev.info/

³GSC, <http://gensc.org/>

1.3 Objective, Goals and Scope of the Thesis

The *openEHR* archetype approach supports the development of future-proof systems and data, domain empowerment, knowledge-level interoperability and intelligent querying [17]. By basing the development of the eBMRR on the *openEHR* archetype approach, these characteristics can be exploited in the domain of biomedical research. Thus, to answer the posed research question, the goal of this thesis is to investigate the feasibility of modelling the information that needs to be recorded in the eBMRR with *openEHR*-based archetypes. The eBMRR needs to support the recording of clinical patient information, information on the patient’s biospecimen and the data produced through research on the patient’s biospecimen.

1.3.1 Objective 1: Reusability of clinical archetypes

Clinical archetype development is an ongoing effort in the *openEHR* community. Thus, one of the objectives of this thesis is to test the reusability of the existing clinical archetypes to model the clinical information in a biobank, using the information encoded in the database of the BIMS of the Irish Prostate Cancer Research Consortium (PCRC) biobank as an example.

1.3.2 Objective 2: Development of omic archetypes

In addition to the clinical data, the eBMRR needs to support the recording of omic research data. Thus, omic concepts need to be modelled with a set of generic and customisable archetypes that can be easily adapted to a specific biobank or study. The development of these archetypes will leverage recently emerging data recording standards in the research domain for recording the execution and results of omic experiments. Used in combination, the clinical and omic archetypes can then be used to construct the proposed eBMRR that can integrate clinical information about a patient/donor with research information won from omic experiments in a generic way to support biomedical research. By basing the proposed eBMRR on the *openEHR* archetype approach, clinical patient information and research information on patient samples may be shared safely and can be reused in future research. Furthermore, the *openEHR* approach involves the development of generic and flexible archetypes, which can be adapted to and reused in different situations, according to specific information requirements, thus promising to significantly ease the process of setting up new biobanks information management systems.

1.4 Research Approach

In order to fulfil the stated objective and goals of this thesis, the following research approach has been taken: Specifically, the objective is to investigate the feasibility of using the *openEHR* archetype approach to construct the eBMRR that integrates both clinical patient and (omic) research information in a biobank setting.

Thus, the task at hand is to model both the clinical patient information and the (omic) research information on patient samples in a biobank with generic archetypes. The steps involved are as follows:

- Research and survey the state of recent EHR approaches, particularly those promoting a two-level modelling approach.
- Research and survey the state of the art of biobanks and bio-repositories.
- Research and survey the state of the art of omic data sharing, including omic databases and omic reporting standards.
- Model the clinical information in a biobank with *openEHR* archetypes.
- Design and evaluate archetypes for omic research data and related concepts in research, such as experimental protocols, based on suitable information recording standards.
- Evaluate the use of the *openEHR* approach for the development of the proposed eBMRR.

1.5 Contribution to State of the Art

This thesis investigates the feasibility of using the *openEHR* archetype approach to create a novel type of record, the eBMRR, that integrates different types of research information, for example omic information created by recent high-throughput technologies, with clinical patient information to support bio-medical research in biobanks. The contributions of this thesis to the state of the art of the archetype approach are as follows:

1. an evaluation of the reusability of clinical archetypes to model clinical data in a biobank setting
2. an approach to modelling data in a biobank system with *openEHR* archetypes
3. new insights into the challenges associated with integrating data in legacy databases into an archetype-based system

4. the definition of a set of archetypes to model the data in a legacy BIMS, namely the BIMS of the Irish Prostate Cancer Research Consortium (PCRC) biobank
5. the definition of a strategy for developing omic archetypes based on existing omic reporting standards
6. the development of sets of generic omic archetypes that model omic information based on existing omic data recording standards
7. new insights into the wider applicability of using the *openEHR* archetype approach to model other information domains outside of the domain of healthcare delivery, such as omic research information

The developed archetypes can be reused and customised easily to different research information storage requirements in a specific biobank or study. This would ease the process of setting up a new biobank or study while enabling the safe sharing and communication and future-proof storage of information to support biomedical research.

1.6 Thesis Overview

This thesis proposes the development of a novel type of record, the electronic Bio-Medical Research Record (eBMRR), based on the *openEHR* archetype approach. This record aims to generically integrate both clinical and research information to support biomedical research. This thesis describes the process of modelling the information in the eBMRR with archetypes. This includes the modelling of both clinical data on sample donors and the data produced through experimental analysis on the patient samples in biobanks with generic archetypes. These generic archetypes can then be customised to satisfy the information needs in a particular biobank or study. An evaluation and demonstration of the use of these archetypes is provided by adapting them to specific situations through the use of templates and mapping them to specific instance data.

This introduction is followed by a chapter (Chapter 2) reviewing the state of the art of several areas important to this thesis. Firstly, the background for the emergence of standards in the domain of clinical data sharing is provided. These standards provide an overview of the attempts to overcome the difficulties of sharing clinical data. The review of these standards developments provides an important basis to understand the issues around sharing clinical data in a large, complex and evolving domain. It also provides a justification for choosing the *openEHR* approach as the basis for the development of the proposed eBMRR and describes relevant parts of the approach in detail. Secondly, a state of the art of biobanks and bio-repositories is provided, to

underline the need for the proposed eBMRR. Finally, the state of the art of sharing omic data is presented, providing an overview of public omic databases and developments of omic data standards to ease the sharing of omic data. This chapter provides the background upon which this thesis is built and positions the proposed eBMRR.

Chapter 3 describes the modelling of the information in a legacy database of a biobank with *openEHR* archetypes. This chapter provides an insight into the information modelling needs of the eBMRR and demonstrates the concept of reusability of clinical archetypes in a biobank context.

Chapter 4 discusses the design of archetypes that generically model omic information. This chapter demonstrates how omic data recording standards may be leveraged in the design of generic omic archetypes.

The penultimate chapter (Chapter 5) provides an evaluation of using the *openEHR* archetype approach to create the eBMRR, comprising the following sub-sections: It provides an evaluation of mapping the BIMS of the PCRC biobank to archetypes, providing an insight into the challenges of modelling a legacy system with archetypes and of the suitability of reusing clinical archetypes in a biobank context. Further, an evaluation of using *openEHR* to model omic information is provided. This part also provides a discussion on basing the development of omic archetypes on omic reporting standards. The high-level structure of the proposed archetype-based eBMRR is also discussed.

The final chapter (Chapter 6) summarises the insights and key contributions of this thesis and proposes possible avenues of future work that may be advanced from this thesis.

A number of appendices are provided for further information as referenced from the main text. In addition, a CD is attached containing the archetypes and templates reused and developed for the PCRC BIMS, the omic archetypes and templates developed and instance files used during the evaluation process.

Chapter 2

State of the Art

2.1 Introduction

This chapter provides reviews of the state of the art of the following areas. Section 2.2 provides the background for the recent emergence of standards for clinical data sharing, such as those of ISO¹, CEN² and HL7³, and other integration efforts, including an in-depth review of the two-level archetype approach pioneered by *openEHR*. This review provides an overview of current attempts to safely share clinical data and justifies why the *openEHR* approach was chosen as the candidate approach for the development of the proposed eBMR. Section 2.2.14 provides a short overview of standardisation efforts on the level of terminologies in the medical domain. Section 2.3 provides a state of the art review of the art of biobanking, including their importance in translational research, the role of the biobank information management system, the recently increased need for biobanks to share information, collaboration initiatives and issues surrounding information harmonisation, which is a pre-requisite for information sharing. Biobanks play an important part in knowledge discovery as they provide the link between the clinical and scientific domains. Finally, the state of the art of sharing omic data is presented in section 2.4, with an emphasis on emerging omic data reporting standards. These standards will give an overview of the data storage requirements for omic data and may be exploited in the development of omic archetypes.

2.2 Standardization and integration efforts for the sharing of clinical data

2.2.1 Emergence of standards for the sharing of clinical data

With the rise of computerisation in health care environments, health care organisations have worked towards the digitalisation of patient records to try and combat the

¹International Organization for Standardization, www.iso.org

²European Committee for Standardization, www.cen.eu

³Health Level 7, www.hl7.org

inherent problems associated with the use of paper-based records. However, without the presence of standards and guidelines in the past, a proliferation of many different electronic health record solutions has resulted. Currently, the large majority of EHRs are proprietary solutions and are not shareable or interoperable beyond or even within a single health organisation or between different applications in the same organisation [24].

The plethora of different solutions becomes apparent in the different names and overlapping definitions used to describe them, such as Electronic Medical Record, Electronic Patient Record, Computerised Patient Record, Electronic Health Care Record, Personal Health Record etc. [24]. The term most widely used today in the health informatics community to describe a shared, longitudinal, interoperable digitalised patient's health record is "Electronic Health Record" (EHR) [1].

The presence of a multitude of non-interoperable, proprietary and hospital specific electronic health/patient record implementations impede quality medical care, especially when patients move between medical care facilities. To tackle the problem of storing and communicating patient information in a safe and reliable way to support healthcare delivery, several standards organisations have developed/are developing standards to facilitate the development of generic, shareable and interoperable EHRs so that patient information can be shared safely across healthcare organisations. These standardisation efforts have resulted in a number of complementary and competing standards, each focussing on (a) different aspect(s) of EHRs, such as continuity of care concepts, EHR architecture or messaging.

A short description of some of the most important standard developments and other integration efforts in the EHR domain is provided in the following paragraphs. In particular, the following standards and integration efforts and some of their relationships will be presented: ISO 18308, EN 13940 (CONTSYS), EN 12967 (HISA), EN 14822 (GPIC), EN 13606, the HL7 group of standards and a number of other relevant standardisation and integration efforts (ASTM, CDISC, IHE, *openEHR*).

Specific focus is given to standards and developments related to the archetype approach, pioneered by *openEHR* and adopted in the EN 13606 standard, which is a novel modelling approach developed to support interoperability in EHR communication.

2.2.2 ISO 18308

ISO 18308:2011, "Health informatics - Requirements for an electronic health record architecture", defines a set of business objectives and requirements for the architecture of EHR systems that process, manage and exchange EHR information. The purpose of these requirements is to ensure that compliant EHRs fulfil the needs of healthcare delivery, are reliable and clinically valid, follow ethical and legal guidelines and requirements,

promote good clinical practice and support reuse of information for different purposes. The standard revises and replaces the earlier ISO/TS 18308, advancing it through new insights and experience won from implementing EHR systems and work carried out on other related EHR standards since, including work carried out on ISO 13606 (described in Section 2.2.10), parts of HL7 (described in Section 2.2.9 and *openEHR* (described in Section 2.2.12).

The standard also provides a definition of the EHR as *“one or more repositories, physically or virtually integrated, of information in computer processable form, relevant to the wellness, health and healthcare of an individual, capable of being stored and communicated securely and of being accessible by multiple authorized users, represented according to a standardized or commonly agreed logical information model. Its primary purpose is the support of life-long, effective, high quality and safe integrated healthcare”*.

2.2.3 CONTSYS (EN 13940)

The CEN standard EN 13940 or CONTSYS [25, 26], is a two-part standard. Part 1 of CONTSYS (EN 13940 “Health informatics - System of concepts to support continuity of care”), “Basic Concepts”, explains key concepts, data and information flows of continuity of healthcare and their inter-relationships to provide a common ontology and terminology of continuity of care concepts [25]. Part 2 of EN 13940, “Health care process and workflow” defines concepts encountered in the healthcare process, focusing on clinical concepts at the point of interaction between the subject of care and healthcare professionals. The purpose of this part of the standard is to enable semantic interoperability with regard to the information about the healthcare process and the related workflow in order to support continuity of care [26].

2.2.4 HISA (EN 12967)

The CEN standard EN 12967 or HISA (EN 12967: “Health Informatics - Service Architecture”) specifies a standard architecture for healthcare information systems. The goal of this standard is to enable the development of modular open systems in the arena of healthcare. The HISA standard describes any generic healthcare information system as an association of diverse applications that interact and co-operate through a middleware layer of common services (see Figure 2.1, from [27]). HISA specifies the external structure of the data that is handled by each service [27]. This standard is complementary to EN 13606, which is described in section 2.2.10.

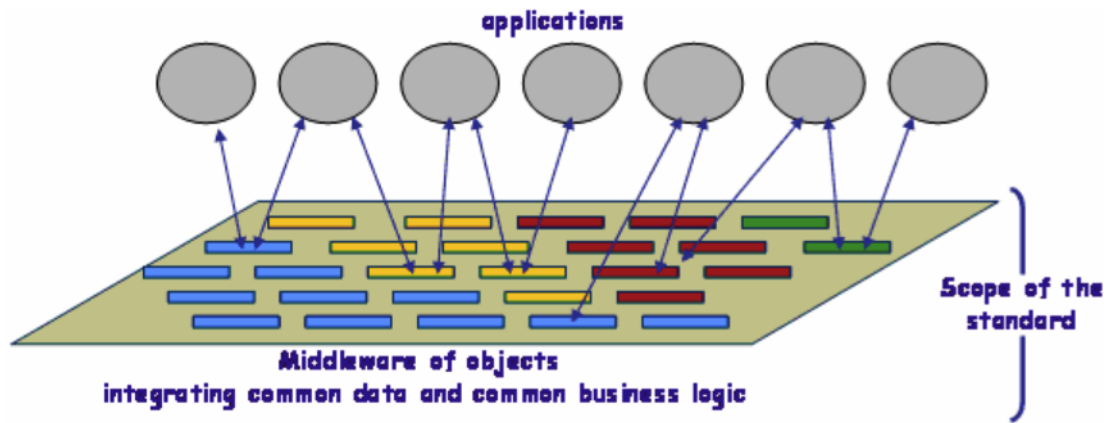


Figure 1 — Scope of this standard

Figure 2.1: Scope of the HISA standard, Source: [27]

2.2.5 GPIC (EN 14822)

The CEN standard EN 14822 or GPIC (EN 14822: “Health informatics - General purpose information components”) is another complementary standard that specifies a set of general-purpose components that are based on objects defined in several other standards. The background for this standard was the realisation that there exists a multitude of partially overlapping standards for messaging and information structures in healthcare, with similar objects being specified in separate standards, making it hard to implement conformant applications. Furthermore, the standard serves the purpose of harmonising information models for health between the USA (mainly HL7) and Europe (CEN/TC 251 defined models, described in section 2.2.10). The GPIC standard specifies a set of components that describe concepts that are commonly encountered in healthcare information systems and in the exchange of data between healthcare information systems [28].

2.2.6 ASTM healthcare standards

With regard to EHRs, ASTM International⁴ developed the Continuity of Care standard “ASTM E2369 - 05 Standard Specification for Continuity of Care Record (CCR)” [29]. The CCR contains a comprehensive data summary of the most pertinent demographic, administrative and clinical information about the healthcare of a patient and can be used as a vehicle to exchange this information with another healthcare provider to support continuity of care. It is supposed to give a snapshot, an overall clinical

⁴ASTM International, previously known as the “American Society for Testing and Materials”, is a standard developing organisation <http://www.astm.org/>

picture about the current and relevant healthcare status of a specific patient, such as information about health insurance, information about the patient's health status, such as medications, allergies, vital signs, reason for transfer or referral etc. that can be sent to another healthcare provider to support continuity of healthcare. CCRs can be paper-based or electronic. ASTM E2369 specifies an XML schema to ensure interchangeability in CCRs in the electronic version of CCRs [29].

Both the ASTM CCR and CEN CONTSYS standards cover continuity of care, but they differ in focus and are thus complementary. CONTSYS defines continuity of care concepts, such as actors, health issues and their management, organisational principles of healthcare, clinical processes and care pathways across healthcare providers. In contrast, the CCR standard defines the summary information that needs to be passed on from one healthcare organisation to the next, such as patient demographics, allergies, insurance cover and diagnoses, to support continuity of care of a patient between healthcare organisations.

2.2.7 CDISC

CDISC (Clinical Data Interchange Standards Consortium)⁵ develops open, vendor-neutral and platform independent standards for “the acquisition, exchange, submission and archive of clinical research data and metadata”⁶. CDISC standard developments include the Study Data Tabulation Model (SDTM) standard for the regulatory submission of Case Report Tabulations, the Standard for Exchange of Non-clinical Data (SEND), the Laboratory Model (LAB) specified for sending clinical laboratory data, including pharmacogenomics data etc. [30].

CDISC collaborates with IHE (see section 2.2.8) in the development of several CDISC standards with the aim of proposing technical solutions that align healthcare with medical research (e.g. the Retrieve From Data-capture (RFD) integration profile). These technical solutions, such as the RFD integration profile developed in the CDISC Healthcare Link project, supporting standardised capture of data relevant to clinical trials at the point of care⁷. CDISC also collaborates with HL7 in the development of standards, for example on the development of the Biomedical Research Integrated Domain Group (BRIDG) Model for protocol-driven research.

CDISC provides standards that bridge between healthcare and medical research, such as the Operational Data Model (ODM) or the Laboratory Data Model (LAB). While these standardisation efforts play an important part towards standardising data in clinical research, their focus is limited to standardising the recording of the fairly

⁵CDISC (Clinical Data Interchange Standards Consortium), www.cdisc.org

⁶<http://www.cdisc.org/about/index.html>

⁷CDISC Healthcare Link Initiative, <http://www.cdisc.org/healthcare-link>

well defined set of data and metadata needed for clinical trials to support the development of medical and biopharmaceutical products, but does not support other types of biomedical research. In addition, CDISC's technical solutions focus on supporting partial reuse of information from the patients EHR and ease of data entry specifically to support clinical trials research at the point of care. CDISC standards thus tackle a problem different to the one that is the focus in this research, which is the development of a research record that integrates biomedical research data from patient samples with the patients clinical background data and thus the CDISC standards were of limited value to this research and did not provide relevant input towards the development of the eBMRR.

2.2.8 IHE

Integrating the Healthcare Enterprise (IHE)⁸ is a not-for-profit international initiative of industry and healthcare professionals whose aim is to improve the sharing of healthcare information between computer systems to accelerate the adoption of EHRs. To achieve this goal, IHE defines technical frameworks to be used as guidance for the implementation of existing messaging and information standards to accomplish specific clinical aims. IHE promotes the use of established standards such as DICOM (a medical imaging standard) and HL7 etc. instead of defining new standards [31]. After defining critical use cases for sharing clinical information, experts create technical specifications, called IHE Integration Profiles, that address the communication between systems based on these use cases. These IHE Profiles are then implemented by industry in healthcare IT systems and tested by IHE at so-called Connectathons - interoperability and IHE conformance testing events organised by IHE in various locations all over the world [32]. The IHE Technical Frameworks, detailed technical documents, are compilations of IHE Profiles and provide guidance for implementation [32]. IHE uses the concepts of Actors and Transactions in defining their Integration Profiles. IHE defines actors as (components) of information systems that handle clinical or operational information in the healthcare organisations. Correspondingly, transactions are defined as interactions between actors that convey the information using message protocols based on standards. Then, to conform with an Integration Profile, vendors just need to implement the specified actor(s) and transactions [31]. For example, the IHE Integration Profile for "Cross-Enterprise Document Sharing" or XDS enables healthcare providers to share clinical records for a patient. It uses a document registry and federated document repositories to create a longitudinal patient record. XDS is neutral with regard to the content of documents and can handle a large set of various document formats, such as

⁸IHE, www.ihe.net/

simple text and formatted text (e.g. HL7 CDAs, see section 2.2.9), DICOM and other images or structured and coded clinical documents.

2.2.9 HL7

Health Level 7 (HL7) is one of a group of American National Standards Institute (ANSI) accredited standards developing organisations concerned with developing standards specifically in healthcare. HL7's most successful standard is a messaging standard that allows the exchange of administrative and clinical data between heterogeneous healthcare applications, HL7 version 2 (HL7v2). HL7v2 is currently the most widely adopted version of HL7 messages in practice. HL7v2's success is mainly due to its flexibility. However, this flexibility has the disadvantage in that interoperability between vendors is hampered and conformance testing is difficult. It was largely to overcome these problems that HL7 began development of HL7 Version 3 (HL7v3), which follows a significantly different approach [33].

The main difference between HL7v3 and its previous versions is the specification of a Reference Information Model (RIM). The RIM is an object model that represents clinical data or domains and specifies a message's lifecycle. The RIM explicitly defines the semantic and lexical relationships between data conveyed in HL7 message fields. The HL7v3 standard is a lot more definite and offers very little optionality, thus supporting improved interoperability and facilitating conformance testing. The model is shared between all the domains and all domains create their messages from this model [33]. HL7v3 also encompasses two further information models: the D-MIM (Domain Message Information Model), which is a refined subset of the RIM for a particular domain, and the R-MIM (Refined Message Information Model). The R-MIM is a subset of the D-MIM, and specifies the information content for a message or set of messages for the corresponding D-MIM domain [33]. Thus, the HL7v3 approach is a multi-model approach.

In November 2000, ANSI approved the first version of HL7's Clinical Document Architecture (CDA), with the second version approved in January 2005. The CDA is part of the HL7v3 standard and specifies an XML architecture for the exchange of clinical documents (e.g. progress notes and discharge summaries). The CDA is encoded in XML, is based on the HL7 RIM using RIM data types and utilises the HL7 coded vocabularies. This gives CDA documents machine-readability as well as human-readability. XML-aware Web browsers can be used to display CDA documents [33]. The CDA documents add persistence to HL7's general messaging standard and can be transferred using HL7 messages [34]. The HL7 CDA standard competes with CEN's EN13606 standard described in the next section. The relationship and potential for competition between HL7 CDA, EN13606 and the *openEHR* specification described in

Section 2.2.12 is discussed in Section 2.2.13.

2.2.10 EN 13606

EN 13606 is a 5-part standard that defines an information architecture “for communicating part or all of the electronic health record (EHR) of a single subject of care”. This standard does not provide a specification for the internal architecture or database structure of EHR systems or their components. To support EHR interoperability, EN 13606 proposes a generalised approach to representing health record data structures in a consistent way. In particular, EN 13606 adopts the so-called Dual Model or two-level modelling approach pioneered by *openEHR* [35]. This approach distinguishes between two models, the Reference Model and the Archetype Model (see section 2.2.11 on Two-level Modelling below for more details). Part 1 of EN 13606 defines the Reference Model, which represents the generic properties of health record information. Part 2 of EN 13606 then defines the Archetype Model to which archetypes, which are formal expressions of distinct, domain-level concepts, conform. Part 3 of EN 13606 provides reference archetypes and term lists of values that may be used by the reference model defined in Part 1. Part 4 of EN 13606 covers security issues, such as access privileges relevant to EHR communication. Finally, Part 5 of EN 13606 provides an interface specification for the exchange for record extracts. The standard has also been approved by ISO. Part 1 of EN 13606 supports, but does not mandate, the use of archetypes.

The Reference Model defined in Part 1 of EN 13606 is based on the requirements defined in ISO 18308 and is a hierarchical information model (the EHR Extract Reference Model) that could be utilised to characterise a message, an object interface or an XML document or schema that can be exchanged between heterogeneous clinical systems while preserving information semantics [36]. The two-level modelling methodology defined in this standard is described in detail in the next section.

2.2.11 The two-level modelling methodology

Traditionally, information systems are developed using a “single-level” methodology. In this approach, business entities are modelled and then implemented directly into software and databases. For example, concepts are often hard-coded directly into objects and/or relational database schemas (see Figure 2.2). This approach has several shortcomings for the development of information systems in large, complex and constantly changing domains, such as healthcare. For example, only currently known requirements are captured - often incompletely, implementing changes is time-consuming, expensive and risky, standardisation and thus interoperability are difficult to achieve [17].

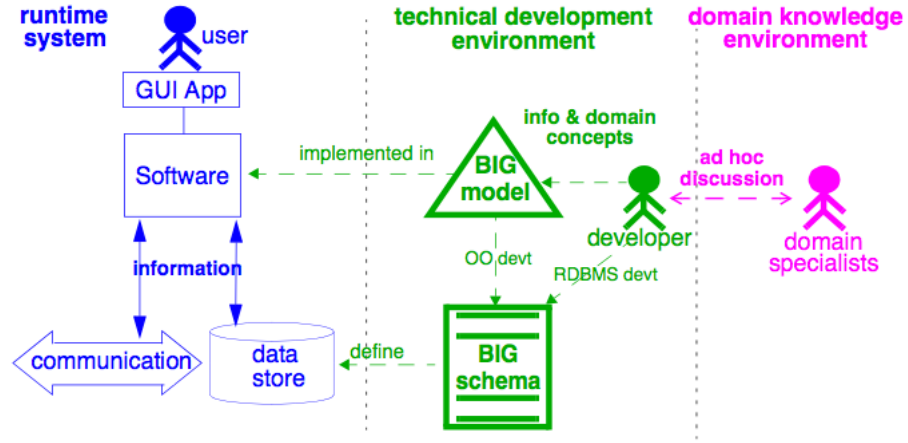


Figure 2.2: Single-level information system development methodology (Source: [17]).

In the two-level modelling methodology pioneered by *openEHR* to model clinical information, a clear distinction is made between the modelling of information and knowledge, which are modelled at two levels [37, 17, 38, 39]. This two-level methodology is graphically depicted in Figure 2.3.

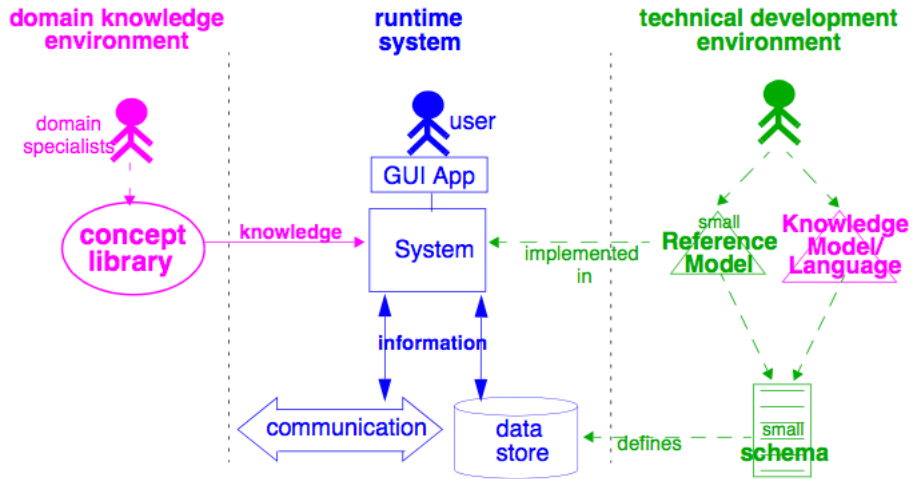


Figure 2.3: Two-level information system development methodology (Source: [17]).

Using ISO/TS 18308 as a reference set of requirements, the first level reference model specifies the global and stable characteristics of the components of the health record, how they can be aggregated/composed and what kind of information needs to be provided to meet requirements of legal, ethical and provenance nature. The first level models the semantics of information, such as generic data types and structures in a small and stable Reference Model [17, 36]. These components are defined in the Reference Model as a set of classes that form the generic information building blocks of the EHR [40].

The Reference Model is hierarchically organised with the EHR Extract acting as top-level container of an EHR of a single subject of care [36] and thus builds the root

of the aggregation hierarchy. The main hierarchy components of the EHR Extract Reference Model are displayed in Figure 2.4 (adapted from [41]).

EHR Extract	Part or all of the electronic health record for one person, being communicated
Folders	High-level organization of the EHR e.g. per episode, per clinical specialty
Compositions	Set of entries comprising a clinical care session or document e.g. test result, letter
Sections	Headings reflecting the flow of information gathering, or organizing data for readability
Entries	Clinical statements about Observations, Evaluations, and Instructions
Clusters	Multipart-entries, tables, time series, e.g. test batteries, blood pressure, blood count
Elements	Element entries: leaf nodes with values e.g. reason for encounter, body weight
Data Values	Data types for instance values e.g. coded terms, measurements with units

Figure 2.4: Hierarchy of the ISO/EN 13606 Reference Model (Source: [41])

In combination with suitable attributes for each level of the EHR Extract hierarchy, these logical building block classes enable the provision of the contextual requirements necessary to ensure faithfulness in EHR communication [36]. The EHR Extract contains Compositions that can optionally be organised in Folders. Similarly, Compositions contain Entries that may be organised in different Sections. Finally, Elements contain Data Values and can be defined in Clusters [36, 41], as shown in Figure 2.5 (from [36]).

In an *openEHR*-based record, the unit of committal, modification, transmission and attestation of information is at the level of the Composition, equivalent to a signed document. The Composition acts as the primary data container and instances of Compositions can be regarded as stand-alone aggregations of data [42].

The second level formally models clinical content in the form of archetypes and templates [17, 39]. Archetypes and templates are used to model domain knowledge, which may be volatile and evolving, for example clinical concepts such as “laboratory report” or “blood pressure measurement”. Archetypes are defined as “constraint-based domain models” or “constraint-based concept definitions” that configure and constrain valid combinations of the information building blocks defined in the Reference Model

to formally represent distinct and complete domain-level concepts with regard to their naming, attribute information, optionality and multiplicity [37, 38, 40].

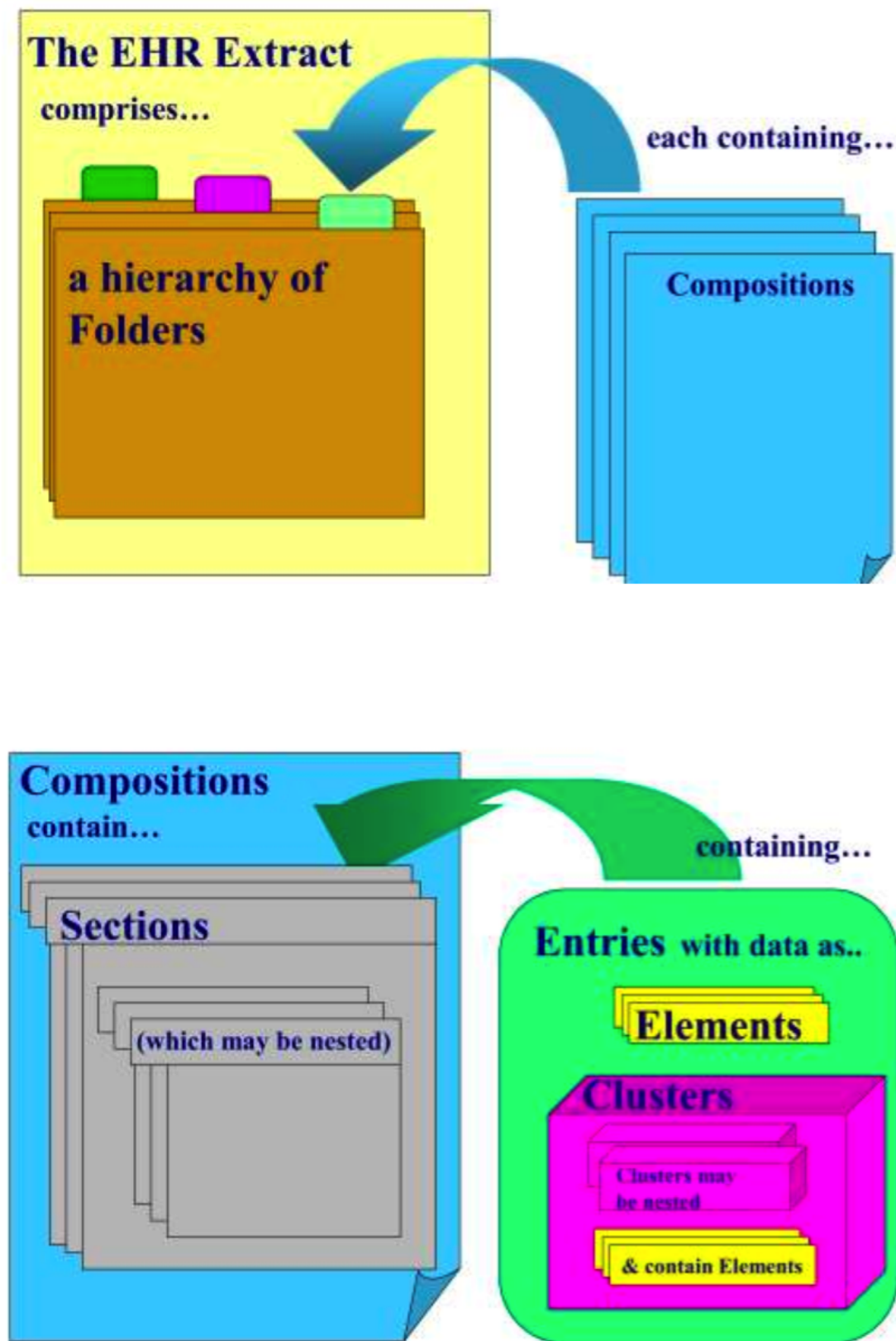


Figure 2.5: Diagram of the EHR Extract hierarchy (Source: [36])

For example, the data contained in the concept “blood pressure” can be formally described as having the elements “Systolic Pressure”, “Diastolic Pressure”, “Mean Arterial Pressure” and “Pulse Pressure”, each representing a quantity (see Figure 2.6). Since an archetype modelling a specific domain concept aims to be complete, i.e. it aims to cover all possible variations of the concept, it should be re-usable [17]. Archetypes can also be composed of lower level archetypes to form other archetypes using a facility called archetype slots [37, 17, 43, 38]. Furthermore, archetypes can be specialised from general concepts into more specific concepts [37].

Blood Pressure (v1)

Archetype: Blood Pressure (openEHR-EHR-OBSERVATION.blood_pressure.v1)

Header Data State Protocol Events

Structure: Tree
Occurrences: 1..1 (mandatory)
Cardinality: 0..* (optional, repeating, unordered)

Q	Systolic Quantity Occurrences: 0..1 (optional) [SNOMED-CT(2003)::163030003]	Peak systemic arterial blood pressure - measured in systolic or contraction phase of the heart cycle.	Property: Pressure Units: • 0.0..<1000.0 mm[Hg] Limit decimal places: 0
Q	Diastolic Quantity Occurrences: 0..1 (optional) [SNOMED-CT(2003)::163031004]	Minimum systemic arterial blood pressure - measured in the diastolic or relaxation phase of the heart cycle.	Property: Pressure Units: • 0.0..<1000.0 mm[Hg] Limit decimal places: 0
Q	Mean Arterial Pressure Quantity Occurrences: 0..1 (optional)	The average arterial pressure that occurs over the entire course of the heart contraction and relaxation cycle.	Property: Pressure Units: • 0.0..<1000.0 mm[Hg] Limit decimal places: 0
Q	Pulse Pressure Quantity Occurrences: 0..1 (optional)	The difference between the systolic and diastolic pressure.	Property: Pressure Units: • 0.0..<1000.0 mm[Hg] Limit decimal places: 0
T	Comment Text Occurrences: 0..1 (optional)	Comment on blood pressure measurement.	Free or coded text

Figure 2.6: Blood pressure archetype (Source: *openEHR* Archetype Repository⁹)

In practice archetypes are ultimately deployed through templates that combine several archetypes to represent screen forms, documents, printed reports or messages, such as “discharge summary” [38, 39] and are usually developed locally to customise archetypes to the local context [44]. For example, default values can be set at a template level and optional archetype items can be deleted or mandated depending on local usage requirements. Data entry and validation usually happens on a template level. These templates “soften the rough edges” inherent in using a small information reference model by adapting archetypes to the situation at hand. An example of a screen from generated from archetypes is shown in Figure 2.7.

The screenshot shows a web-based form for 'Objective_Examination' within the 'openEHR' and 'UCL' interface. The form is organized into three main sections: a left sidebar with navigation links, a central data entry area, and a right sidebar with patient records and reports.

Left Sidebar (Observations):

- Clerking
- History
- Examinations
- Investigations
- Test Archetype (Connected to Reporting)

Central Data Entry Area:

- Objective_Examination: General appearance (text input)
- Vitals signs:
 - Respiratory Rate (text input)
 - Heart Rate (text input: 100.0)
 - Blood Pressure Systolic (text input)
 - Blood Pressure Diastolic (text input)
 - Oxygen Sats % (text input)
- Structured Scales:
 - Glasgow Coma Score (text input)
 - Early Warning Score (text input)
 - Mental State (text input)
- Head & Neck Exam:
 - Head & Neck exam (text input: Normal appearance)
 - other H&N detail (text input)
- Chest Exam:
 - Chest Exam (text input)
- Cardiovascular system:
 - Heart sounds I & II norma (text input)
- Respiratory system:
 - Normal examination (text input)
- other CVS & Resp examination detail (text input)
- Abdominal/GenitoUrinary Exam:
 - Abdominal examination (text input: soft non tender abdomen)
 - other Abdominal examination detail (text input)
- GenitoUrinary exam (text input)
- Nervous system:
 - Cranial Nerves (text input: Cranial Nerves normal)
 - Peripheral Nerves (text input: Upper Limb TPCRS norm)
 - other CNS detail (text input)
- Musculoskeletal system:
 - Upper Limbs (text input: Shoulder/Elbows/Wrists/F)
 - Lower Limbs (text input: Hips/Knees/Ankles/Foot r)
 - other Limb detail (text input)
- Skin (text input)

Right Sidebar (Patient Records):

- Update
- Objective_Examination
- General appearance :
- Vitals signs
- Respiratory Rate :
- Heart Rate : 100.0 U
- Blood Pressure Systolic :
- Blood Pressure Diastolic :
- Oxygen Sats % :
- Structured Scales
- Glasgow Coma Score :
- Early Warning Score :
- Mental State :
- Head & Neck Exam
- Head & Neck exam : Normal appearance
- other H&N detail :

Buttons: 'Update' button at the bottom right.

Figure 2.7: Screenshot of archetype generated screen form (Source: Opereffa online demo¹⁰)

A template hierarchically combines and constrains archetypes to build complete Compositions in an EHR that can be used to capture application-level lumps of information [39]. The relationship between archetypes and templates is shown in Figure 2.8.

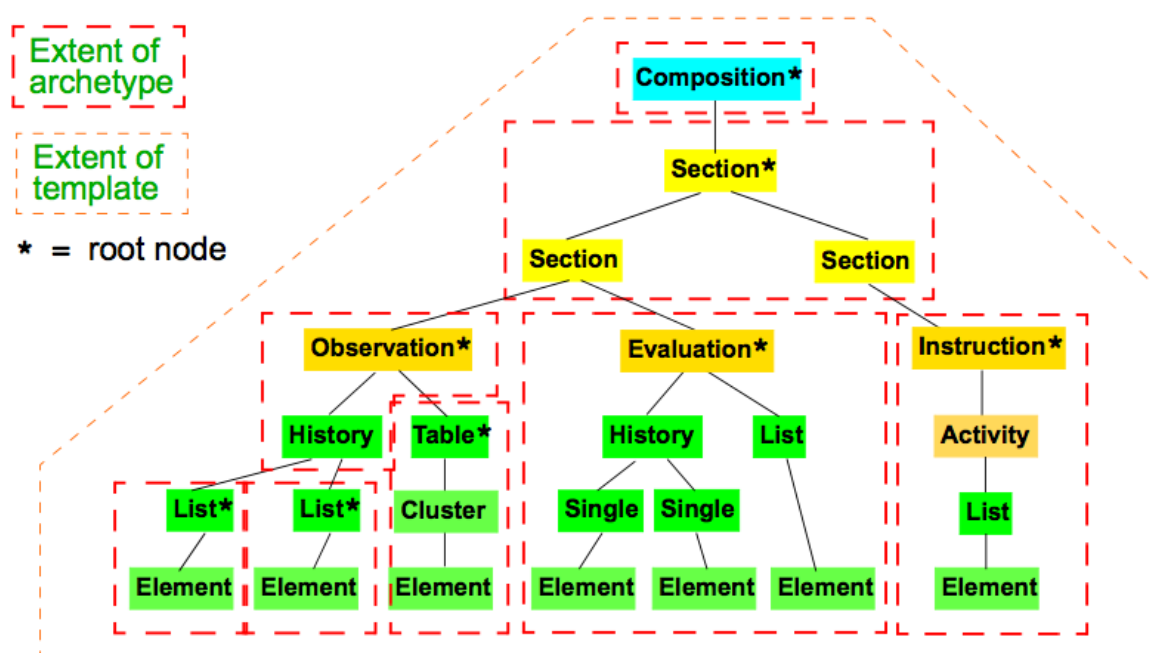


Figure 2.8: Relationship between archetypes and templates (Source: [39])

Only the first level, i.e. the Reference Model, is concretely implemented in software objects and database schemas, whereas archetypes and templates are used at runtime by the system [17]. The consequences of this methodology are that it allows for the development of flexible and future-proof systems [37, 17, 39]. This is important in healthcare systems which are characterised by the large number of concepts, complexity and a high rate of change [37, 17, 39]. Further, the methodology promotes a separation of technical concerns and domain concepts in information systems, meaning that domain experts can formally define domain concepts without technical knowledge of the software of the information system that will process them, thus empowering the user [37]. Further, the two-level methodology enables knowledge-level interoperability and intelligent querying [17].

The two-level modelling approach was adopted in the definition of the CEN EN 13606 standard, described in Section 2.2.10.

2.2.12 *openEHR*

openEHR is a not-for-profit Foundation, founded in 1999 by University College London, UK, and Ocean Informatics pty., Australia which aims to realise interoperable, life-long electronic health records and improve healthcare. *openEHR* develops open specifications, software and knowledge resources, participates in clinical implementation initiatives, collaborates in the development of international health informatics standards and supports health informatics education [45]. The work of *openEHR* has influenced international standards developments in the area of electronic health records, including ISO, HL7 and CEN standards [46].

openEHR has pioneered the Dual Model approach (see section 2.2.11) which has been subsequently adopted by CEN and ISO in the EN 13606 standard as described in section 2.2.10. The *openEHR* specification also includes a specification of an EHR system and provides a full specification for the exchange of EHR Extracts and for the creation, maintenance, storage and querying of EHRs [35]. In addition, *openEHR* defines an EHR reference model, which is a model of an interoperable EHR, defining a logical EHR information architecture [42]. The *openEHR* architecture covers security and confidentiality, including access control, audit trails, anonymity, privacy, integrity, versioning and change management [39]. The *openEHR* approach conforms with ISO 18308. The relationship between *openEHR*, EN13606 and HL7 is discussed in detail in section 2.2.13.

The *openEHR* architecture consists of the following abstract specifications: the Reference Model (RM), the Archetype Model (AM) and the Service Model (SM) [39]. The RM specifies the components of the first level in the dual-modelling approach and how they can be combined, modelling data types and data structures, EHR Extract, demo-

graphics, security, EHR information etc. The AM specifies the semantics of archetypes and templates, including a syntax specification (the Archetype Definition Language, ADL) to create textual archetypes, and how they are used in *openEHR*. Finally, the SM provides a specification for fundamental services in a health information environment, such as a Terminology Interface Model, an EHR Service Model, an Archetype Service Model etc [39].

While the specifics of the *openEHR* reference model can be found in the *openEHR* specifications available online¹¹, the key concepts relevant to this thesis are introduced below. The *openEHR* Foundation continuously work on defining new and enhanced *openEHR* specifications, which can be found on the *openEHR* website¹².

As mentioned in Section 2.2.11, archetypes are formal (computable) constraint-based models of domain concepts. In other words, an archetype formally models some domain concept, expressed as constraints on an underlying reference model [17, 43]. Archetypes can be used to control the the creation, modification and querying of the structure of data and content [43]. The semantics of archetypes (and templates) and their use within *openEHR* are defined in the *openEHR* Archetype Model (AM) [39]. The AM consists of three parts: the Archetype Object Model (AOM), the Archetype Definition Language (ADL), and the *openEHR* Archetype Profile (oAP) [43]. These will now be described in more detail.

Archetype Object Model

Archetypes can be modelled as objects, possibly by parsing a syntax expression of an archetype (e.g. expressed in ADL, XML or OWL) into an object, creating an object parse tree [43]. In particular, an archetype is modelled as a particular kind of AUTHORED_RESOURCE, and includes descriptive meta-data, language information and revision history, identifying information, a definition section and an ontology section [43], as shown in Figure 2.10. The heart of an archetype is the archetype definition. It defines the structure of the archetype and therefore how the information should be recorded. The structure of an archetype definition hierarchically alternates constraints on objects and attributes, building the basis for using paths to reference any node in an archetype [43]. The ontology section adds meaning to the archetype definition section by providing term definitions which are linked to the object nodes in the archetype definition section through the archetype node IDs. This ontology section provides the necessary meta- and contextual information that allow for the safe sharing of information between organisations and the ability to reason about the

¹¹ *openEHR* specifications, <http://www.openehr.org/releases/>

¹² *openEHR* specifications, <http://www.openehr.org/svn/specification/TRUNK/publishing/roadmap.html>

models and intelligent querying [17]. Archetypes themselves are terminology-neutral, but mappings between terms used in the archetype and external terminologies can be established in the ontology section [43, 39]. A number of papers have been published researching the association of archetype terms with medical terminologies [47, 48].

The structure of the definition section will now be described in more detail. The definition section of an archetype essentially consists of a tree of nodes representing alternating constraints on complex objects (defined as C_COMPLEX_OBJECT) and constraints on attributes (defined as C_ATTRIBUTE with subtypes C_SINGLE_ATTRIBUTE for constraints on single-valued attributes and C_MULTIPLE_ATTRIBUTE for constraints on multiple-valued attributes), with constraints on primitive object types (such as a constraint on a String or Integer) at the leaves (defined as C_PRIMITIVE_OBJECT). Both provide constraints on whether the attribute they constrain must exist in an instance file [43]. The difference between C_SINGLE_ATTRIBUTE and C_MULTIPLE_ATTRIBUTE is the objects constrained by the children of the C_SINGLE_ATTRIBUTE define instance alternatives, whereas the objects constrained by the children of the C_MULTIPLE_ATTRIBUTE can co-exist in an archetype instance file and C_MULTIPLE_ATTRIBUTE provides constraints on cardinality (ordering and uniqueness) of the container [43], as shown in Figure 2.9.

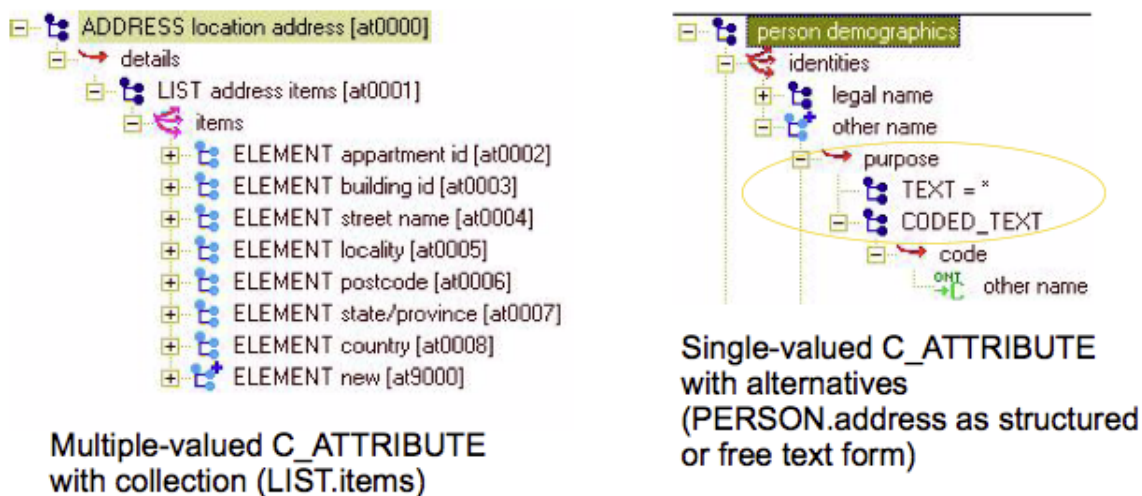


Figure 2.9: Single and Multiple-valued C_ATTRIBUTES (Source: The *openEHR* Archetype Object Model [43])

The root of the archetype definition is an instance of a C_COMPLEX_OBJECT. It expresses constraints on an object of a reference model object type (e.g. allowed occurrences in an archetype instance) and defines what configurations of reference model class instances conform to the archetype [43]. Constraints can be in the form of the type of reference model object chosen, optionality, cardinality, uniqueness etc [43]. At the leaf nodes of an archetype definition the nodes are in form of C_PRIMITIVE_OB

ECT, which define constraints on the value of objects, such as Integers, Strings etc. C_COMPLEX_OBJECT and C_PRIMITIVE_OBJECT are subtypes of C_OBJECT [43]. Simplified, an archetype definition forms a regular cascade of C_COMPLEX_OBJECT/C_ATTRIBUTE/.../.../C_PRIMITIVE_OBJECT, with a C_COMPLEX_OBJECT at the root of the definition, with children of type C_ATTRIBUTE constraining the attributes of the reference model type defined by the parent C_COMPLEX_OBJECT. Every C_ATTRIBUTE has children of type C_COMPLEX_OBJECT constraining further objects etc. and this structure is repeated until there is a constraint of type C_PRIMITIVE_OBJECT on a leaf object [43]. There exist several domain-specific extensions for leaf types, such as C_QUANTITY, which have different constraint semantics from the default C_COMPLEX_OBJECT/ C_ATTRIBUTE cascade [43]. These are defined in the *openEHR* Archetype Profile [49]. Additionally, there are nodes that represent internal references to other nodes in the archetype definition (defined as *Archetype.internal_ref*), nodes that refer to text constraints (*Constraint_ref*) and nodes that represent constraints on the use of other archetypes at that location (defined as *Archetype.slot*) [43]. This hierarchical structure of alternating object and attribute constraints builds the basis the creation of paths that can be used to uniquely identify any node in an archetype. The path syntax used by *openEHR* is similar to W3C Xpath [43]. The exact path syntax is described in [50].

ADL: The Archetype Description Language

ADL syntax is a possible serialisation of an archetype (XML is a possible alternative and the archetypes can be easily converted from ADL to XML syntax and vice versa). ADL provides an abstract syntax for expressing archetypes as structured text files [43]. For example, a constraint on occurrences of an object node in the data is expressed as “occurrences matches {x..y}”, where {x..y} specifies the interval of allowed occurrences. Thus, to express that an ELEMENT is allowed to occur zero or one times in the data, the ADL syntax would be as follows: “ELEMENT[at0003] occurrences matches {0..1}”, where [at0003] is the node identifier that is defined in the ontology section of the archetype as a term code.

The model of an archetype is mirrored in the structure of the ADL document [43], as shown in 2.10.

Archetypes are defined using both dADL (data ADL) and cADL (constraint ADL). dADL is used to formally describe instance data based on an underlying information model, and is both human and machine readable. cADL is used to express constraints on data defined by object-oriented information models. The *openEHR* specification for the Archetype Definition Language describes the ADL syntax in detail [50].

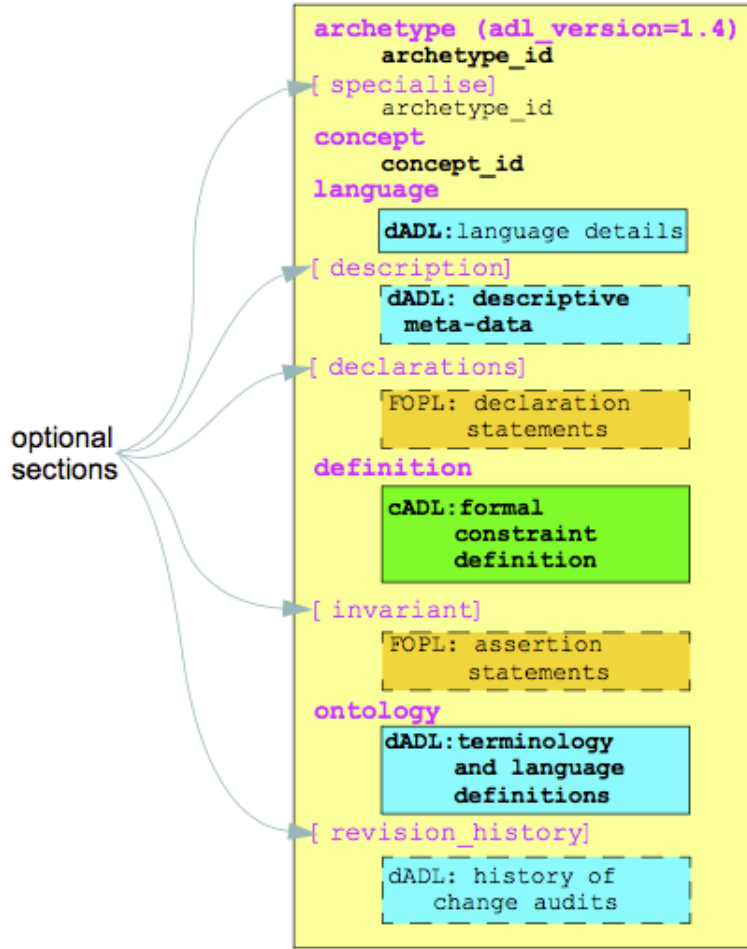


Figure 2.10: ADL Archetype Structure (Source: [50])

An archetype query language (AQL) specification for querying archetype-based EHRs is currently being developed by *openEHR*, based on initial efforts by the Ocean-Informatics company [51].

Recently, the use of the Ontology Web Language (OWL) instead of ADL for the specification of clinical archetypes has been proposed to enable semantic activities, such as semantic reasoning and to allow existing semantic support tools, such as tools to compare, select, classify and perform consistency checking to be used [52]. Combining *openEHR* archetypes with ontologies can allow for reasoning [53] and archetype validation [54].

Archetype development and domain knowledge governance

Archetypes covering specific concepts should be non-overlapping, i.e. they should describe whole and discrete concepts, and be generic, so that they are reusable in different clinical environments [44, 38]. To achieve wide applicability and reusability, ideally archetypes should be developed in a consensus-based approach, involving collaboration

of domain experts from different areas.

The development of *openEHR* archetypes to model clinical content is an ongoing activity within *openEHR* and they provide an online archetype repository of developed archetypes¹³. Another organisation active in archetype development is the UK National Health Service (NHS). The NHS has been developing archetypes and templates as part of a pilot study for the NHS Connecting for Health initiative to test the applicability of CEN13606 as an EHR standard, resulting in nearly 250 *openEHR* archetypes and 40 templates in the pilot study with further work resulting in a total of about 650 archetypes and 60 templates. NHS has provided these archetypes to one of its clinical application developers [55, 56]. The NHS archetypes are available in an online repository¹⁴.

Development of mainly *openEHR* archetypes is also ongoing in the community at large, with initiatives generally focusing on archetype development for domains specific EHRs, such as neonatal care [57], complementary and alternative medicine [41], and archetypes supporting shared home-care for the elderly [58]. Others are focussing on authoring archetypes for the development of guideline or work-flow support and decision support systems in specific areas, such as stroke [59] and chronic disease management [60, 61], or on administrative concepts, such as healthcare authorisation and claim submissions [62].

The *openEHR* archetype approach is still fairly new and several issues have been identified, such as insufficient or immature tool support [57, 63], archetype model limitations [64] and the difficulty of involving clinicians in archetype design [65]. Researchers have also realised the need for international collaboration on archetype development [66] and a number of archetype development processes and domain knowledge governance approaches to control the development of new archetypes have been proposed [67, 68]. For instance, Garde et al. [69] discuss the need of Domain Knowledge Governance for *openEHR* archetypes to avoid ‘rank growth’ of archetypes that would jeopardise semantic interoperability. They call for systematic organisation of archetypes development and maintenance. Kohl et al. [68] provide a possible model by which the distributed but cooperative development of archetypes can be facilitated and governed. Recently, quality requirements for archetypes have been proposed [70].

The archetype modelling approach has also been applied to support data modelling for purposes other than EHR development. For example *openEHR*-based archetypes have been developed to model clinical data sets (CDSs), as a tool to overcome the shortcomings of CDSs, such as conflicting basic data types and to provide a uniform model that helps to better structure CDSs. In this case the archetype methodology is

¹³Clinical Knowledge Manager, *openEHR* archetype repository, www.openEHR.org/knowledge/

¹⁴UK NHS archetype repository, <https://svn.connectingforhealth.nhs.uk/svn/public/nhscontentmodels/>

used to enable the better modelling of new CDSs with archetypes acting as a common basis of discussion [63]. Similarly, *openEHR* archetypes have been modelled to document and govern the nursing profession's domain knowledge to empower multi-centre research [71].

Tools and implementations

Several groups are working on archetype tool support. For example, Sundvall et al. [72] developed an archetype editor that supports the manual or semi-automated creation of bindings between archetypes and terminology systems. Another group reports on the release of LinkEHR-Ed, which is a visual tool for developing archetypes and to create mappings from legacy data sources, such as relational databases and XML data sources to integration archetypes to provide semantic layer [73]. Other support tools are provided by the company Ocean Informatics, including the free Ocean Informatics [74] archetype editor¹⁵ and the Ocean Informatics Template Designer¹⁶ (license required). In addition, an open-source archetype editor has been developed by the Linköping university¹⁷ [75] and more modelling tools are becoming available, such as those provided by the LinkEHR Normalization platform¹⁸. LinkEHR includes a multi-model archetype editor that supports several information models, including CEN EN 13606 and *openEHR*.

With regard to implementing an *openEHR*-based system, several software projects exist. For example, a Java [76] and a Python Reference Implementation of the *openEHR* Reference Model have been developed. Several prototype implementation efforts for *openEHR*-based systems have been carried out. For example, the Clinical Review Board of *openEHR* is working on a project, Opereffa¹⁹, for creating an open source clinical application, built on top of the Java based open source framework, which is using the existing open source Java reference implementation of *openEHR* [77]. Opereffa has been reused in other implementation efforts [78]. Other reports of implementation efforts include [79, 80], with a number of research initiatives focussing on alternatives for GUI development [64, 81, 82].

¹⁵OceanInformaticsArchetypeEditor, www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-archetype-editor.html

¹⁶Ocean Informatics Template Designer, www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-template-designer.html

¹⁷LiU Archetype Editor, www.imt.liu.se/mi/ehr/tools/

¹⁸LinkEHR, <http://www.linkehr.com/>

¹⁹Opereffa, <http://opereffa.chime.ucl.ac.uk/>

Integration of legacy databases

Research is also ongoing on what approach would be best to upgrade legacy systems to the archetype approach. For example, Moner et al. [83] show the definition of ad hoc ISO 13606 archetypes based on legacy Discharge Reports, using LinkEHR to create the mappings and providing a semantic layer over the integrated information. Chen et al. [84] report on their approach to automatically convert the content model of a regional EHR into *openEHR* archetype format and vice versa, preserving most of the structure and semantics of the original models. However, while this approach allows the underlying format to be expressed in archetype format, the resulting archetypes directly correspond to the data model of the clinical source systems (thus representing “integration archetypes”, which do not model data according to good archetype modelling principles, but rather only model the data in the same format). This pushes the data interoperability issues to the archetype level, rather than solving them [85]. It has been shown that it is a non-trivial issue to upgrade a legacy system to an archetype-based system. Issues relating to this have been identified as early as 2003 by Bird et al. [85] who describe their practical experiences won from two field trials of exporting data from legacy systems into an archetype-based system. Several issues were highlighted. For example, the source system may not always provide data that is mandatory in the archetypes, so that this information needs to be inferred, such as units for quantities, scientific properties, person recording data. Problems were also encountered when the data in the source system was stored in less structured form, such as free-form text fields and the archetype provided structured fields. Another issue that added to the complexity of fitting in the legacy data into an archetype was when different ways of grouping data in target and source formats were used. Additional problems concerned inappropriate or wrongly placed data in the source database. They conclude that the rich functionality enabled by using the archetype approach, such as decision support systems and querying, cannot be exploited if the source systems do not provide the data in the appropriate form.

2.2.13 Relationship between EN 13606, *openEHR* and HL7v3

The Reference Models (RMs) for EN 13606 and *openEHR* are very similar and their classes align closely. However, the scope of *openEHR* is broader than that of EN 13606 [35]. EN 13606 can be regarded as representing a subset of the *openEHR* specification, focusing on the communication of EHR Extracts between EHR systems, whereas the *openEHR* specification includes specifications needed for the implementation of a complete EHR system, such as those covering creation of EHRs, security, audit trails, versioning and change management, amongst others [35, 39], as detailed above.

In contrast, the HL7v3 RIM, “the closest HL7 artefact to an information model” [42], while providing similar data types and some related semantics as the EN 13606/*openEHR* RMs, differs substantially from the EN 13606/*openEHR* RMs. In particular, HL7v3 message definitions are derived by a custom process, namely “refinement by restriction”, from the RIM. This means that the data in messages are not instances of HL7v3 RIM classes, as would be the case in systems based on the EN 13606/*openEHR* RMs [42]. However, active collaborations are ongoing between CEN, HL7 and *openEHR* to harmonise the different standards and to map between the different models [86, 35]. While the HL7v3 RIM and the EN 13606/*openEHR* RMs are too different to enable a mapping, parts of the HL7 CDA can be mapped to a EN 13606/*openEHR* Composition [42]. In particular, the HL7 CDA can be regarded as an approximate subset of the EN 13606 EHR Extract, representing one version of one EHR Extract Composition [35]. The relationship between *openEHR*, CEN13606 and HL7v3 RIM and CDA is represented diagrammatically in Figure 2.11 (from [35]).

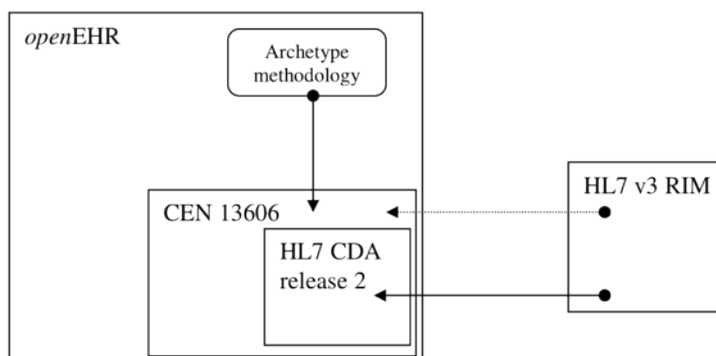


Figure 2.11: Schematic relationship between *openEHR*, EN 13606, and HL7 CDA (Source: [35]).

While the HL7v3 RIM and CDA approach does resemble the EN 13606/*openEHR* two-level modelling approach, HL7 does not provide an equivalent to EN 13606/*openEHR* archetypes (described in section 2.2.11). In addition, HL7 does not currently support archetypes, but collaboration between CEN, HL7 and *openEHR* is ongoing to overcome this.

Several studies compare how specific domain concepts are modelled by different two-level modelling approaches [87, 88], finding advantages and disadvantages in both.

The existence of slightly differing flavours of the two-level modelling approach (CEN EN13606, *openEHR*, HL7 CDA RIMv3) and the availability of competing standards and differing terminologies for annotation (e.g. SNOMED-CT MeSH, NCI, FMA etc.) pose new interoperability issues. Several approaches are being proposed to deal with this issue. For example, Bisbal and Berry [89] present an approach to map archetypes from different sources using ontological alignment techniques. On the other hand, a

method for transforming archetypes between the CEN/ISO 13606 standard and the *openEHR* specification, based on Model-Driven Engineering and Semantic Web technologies by creating ontologies for both standards and a common, integrated archetype model ontology is being proposed [90]. Other research on achieving interoperability between *openEHR*, EN 13606 and HL7 include [91, 92, 93].

Another, more recent approach to generically model information has been the Dutch initiative of detailed clinical models (DCM)²⁰ [94] which can be used to promote interoperability between existing 2-level modelling approaches, such as HL7, *openEHR* and CEN13606 [95, 96].

2.2.14 Controlled vocabularies, terminologies and ontologies in the clinical domain

Controlled vocabularies, terminologies and ontologies play an important part in recording healthcare organisation and are used extensively in EHRs. The standardisation of medical vocabularies supports knowledge management, integration of data and for decision support [97]. Major standardisation efforts of medical terms are briefly introduced below:

- **SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms)**²¹ is a clinical healthcare terminology, managed by the International Health Terminology Standard Development Organisation (IHTSDO). SNOMED CT comprises more than 311,000 active and unique concepts that are formally organised into hierarchies. Concepts are assigned a unique concept identifier numerical code and are linked to human-readable descriptions. In addition to the hierarchical “is a” relationships, SNOMED CT provides other types of relationships, such as “causative agent” or “finding site”.
- **LOINC (Logical Observation Identifiers Names and Codes)**²² provides a coding system for laboratory observations, including entries related to chemistry, microbiology and haematology, and for clinical observations, including entries for cardiac echo, survey instruments, vital signs. It is managed by the Regenstrief Institute and contains over 46,000 concepts [97].
- **ICD (International Classification of Diseases)**²³ is a WHO endorsed classification system for diseases and other health related data, such as injuries,

²⁰DCM, www.detailedclinicalmodels.nl

²¹SNOMED CT, <http://www.ihtsdo.org/snomed-ct/>

²²LOINC, <http://loinc.org/>

²³ICD, <http://www.who.int/classifications/icd/en/>

symptoms, disease causes, reasons for encounter, factors that influence health status etc. It uses alphanumerical codes to assign the classifications.

- **MeSH (Medical Subject Headings)**²⁴ is a hierarchical thesaurus managed by the National Library of Medicine's for categorising medical domains, and is used, for example, for indexing articles in the PubMed database.
- **UMLS (Unified Medical Language System)**²⁵ combines over 100 independent existing medical vocabularies, including LOINC, ICD and SNOMED CT terms, into a combined vocabulary, the UMLS Metathesaurus, by identifying and retrospectively establishing a mapping between synonymous terms [98, 97]. The UMLS provides a Semantic Network that groups concepts and defines semantic relationships between concepts, such as an "is a" relationship.

2.2.15 Summary of EHR standard developments

A review of the most relevant standards to tackle the problem of storing and communicating patient information in a safe and reliable way to support healthcare delivery has been given. Several complementary and competing standards exist. One of the most promising developments towards the development of future-proof, semantically interoperable electronic health records, is the dual model approach, pioneered by *openEHR*, adopted by CEN, approved by ISO and with the HL7 also converging towards this methodology through its CDA specification that is based on the HL7v3 RIM.

In this research, it is proposed that these standard developments in the EHR domain may be exploited for the development of a generic and interoperable electronic biomedical research record, to enable the storage and sharing of clinical patient and research information on patient samples in a safe and reliable way to support biomedical knowledge discovery between collaborating biobanks. Like healthcare, research is a field that is large, complex, rapidly expanding and constantly changing. Therefore, it has been hypothesised that a dual modelling approach like the *openEHR* approach would be particularly suitable to create the eBMRR because of its advantages over the single-level approach in modelling domain specific concepts. Furthermore, basing the development of the eBMRR on an EHR standard will enable the generic integration of research results with archetype-based electronic health records.

There are currently three related efforts that are based on the dual-modelling approach: HL7, EN 13606 and *openEHR*; their relationship has been explained above. In this research, *openEHR*/EN 13606 present a more suitable approach for the development of the eBMRR than the HL7 CDA, due to the characteristics of archetypes, in

²⁴MeSH, <http://www.nlm.nih.gov/mesh/>

²⁵UMLS, <http://www.nlm.nih.gov/research/umls/>

particular their potential for reusability and customisability through templates. Of the three available dual modelling approaches, *openEHR* provides the most comprehensive specification for an interoperable electronic health record architecture, including an EHR Reference Model, which neither EN 13606 nor the HL7 RIM provide [35]. On top of a specification of the communication of EHRs, *openEHR* also provides a full specification for the creation, storage, maintenance, and querying of EHRs [35]. Thus, the *openEHR* specification is proposed as basis for the development of the electronic biomedical research record in this research. Other reasons for choosing the *openEHR* specification are that archetype development is actively being pursued by *openEHR* members, providing an online archetype repository²⁶, the existence of a Java Reference Implementation of the *openEHR* Reference Model [76] and the provision of modelling support tools, such as the Linköping [75] archetype editor²⁷, the Ocean Informatics [74] archetype editor²⁸ and the Ocean Informatics Template Designer²⁹.

2.3 Biobanking

2.3.1 Introduction

To support biomedical knowledge discovery, clinical patient information found in medical records, such as medical history and lifestyle information, needs to be augmented with results from molecular research, such as genomic, proteomic and other “omic” data produced by novel high-throughput technologies on human biospecimens. Biobanks, or bio-repositories, play a central role in combining these two streams of information [4]. Biobanks are places that collect, store and distribute biological samples, such as blood, urine and tissue and their derivatives for diagnostic and/or research purposes [15]. In addition, biobanks also store associated clinical patient data, such as medical history, lifestyle and outcome data and may also store data from research experiments or analyses on the specimen from the biobank. Thus they combine clinical and lifestyle information of patients with research results, which is essential to elucidating pathological disease processes [8].

Biobanking has been carried out in many institutions for a long time, in a rather informal way [10]. The roots of biobanking can be found in clinical pathology, but biobanking today is a young industry and can be regarded as its own research discipline with many specialised components, spanning frameworks, equipment, operating proce-

²⁶Clinical Knowledge Manager, *openEHR* archetype repository, www.openEHR.org/knowledge/

²⁷LiU Archetype Editor, www.imt.liu.se/mi/ehr/tools/

²⁸Ocean Informatics Archetype Editor, www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-archetype-editor.html

²⁹Ocean Informatics Template Designer, www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-template-designer.html

dures, clinical informatics, database structures, policies, economic models, governance models and specialised personnel [15, 99]. The concept of biobanking is heterogeneous [100, 10]. Biobanks differ in size, ranging from small disease-specific collections of biospecimens, such as the Baker IDI biobank for investigating cardiovascular disease³⁰ [11] or prostate cancer specific biobanks, such as Procure³¹, to large population-based biobanks, such as the Estonia Genome Project³² or the UK Biobank³³ or Iceland’s DeCode Biobank³⁴, by purpose, e.g. those that mainly support clinical healthcare, such as pathology archives for medical diagnosis versus those that have been set primarily for research purposes, and by the biological material collected [10, 15].

Recent advances in biotechnology, such as the emergence of high-throughput technologies, have increased the current demand for high-quality, well-annotated human biospecimens in biomedical research [7, 9, 6, 8, 100, 5]. Because of the increased demand, new biobanks are being created all over the world and biobanking activity is increasing [10]. The establishment of collections is often motivated by specific diseases and small sample collections usually outnumber larger ones [10]. The purpose of most current biobanks is sample management, but there is now an increased drive to incorporate research data from experiments into the biobanks as well [14].

2.3.2 The Biobank Information Management System

Human biospecimens play a vital role in translational and epidemiological research and personalised medicine, for example in the discovery of risk factors, disease biomarkers or biomarkers for personalised drug development and targeted therapies [101, 102, 103, 104, 105, 106, 107, 108, 13, 109]. However, to be of value for research, the biological material must be associated with an accurate collection of detailed phenotypic/clinical patient information, such as the medical history of the sample donor, since it is only then that correlations can be drawn about the influence of genes and/or the environment on disease pathology [12, 11, 14, 13]. As a consequence of the advances in information technology, there has been an almost exponential increase in the information created in biomedical research, which needs to be managed [14].

The data in biobanks is managed by the biobank information management system (BIMS). The BIMS stores information about samples (sample annotation) and the clinical background of the patient from which the sample has been taken, together with sample administrative data. Furthermore, biospecimens are annotated with regard to the composition of the sample and a link to the clinical patient information, such as

³⁰The Baker IDI Biobank, http://www.bakeridi.edu.au/core_facilities/biobank/

³¹Procure, www.procure.ca

³²The Estonia Genome Project, www.geenivaramu.ee/

³³The UK Biobank, www.ukbiobank.ac.uk

³⁴DeCode, www.decode.com/

the disease, treatment and patient outcome is established [15]. Clinical information that is pertinent to the research being carried out is generally manually extracted and imported into the BIMS from the patient’s health record and/or from questionnaires and/or interviews with the patient/donor. However, reflecting the heterogeneity of biobanks, current BIMS solutions tend to be bespoke disease- and study-specific proprietary implementations, such as the BIMS developed by Olund, Lindqvist, and Litton [14] extensively using IBM specific off-the-shelf components, or the database developed by [13] using Velos Eresearch software (Fremont, California), based on XML, Enterprise Java Beans and an Oracle database. While these two implementations represent quite sophisticated solutions that are able to integrate data of different formats and from different data collection methods from different research studies, other BIMS solutions may be as simple and archaic as text files and spreadsheets maintained by a biobank administrator with research results scattered over individual researchers’ computers. In addition, there is often no provision to feed back research results that may have been expensive and time-consuming to produce into the BIMS, so that they are kept scattered over individual researchers’ computers, often lost when the researcher leaves the institute, and thus unavailable for further analysis or comparison. However, more and more researchers believe that there is value in storing this research data, together with the clinical data, for future research and analyses. In fact, some researchers believe that we are now on the verge of being able to combine the results of molecular analyses retrieved with high-throughput genotyping technologies with life-style and demographic data on a large scale [14], requiring the storage of today’s research results for tomorrow.

Major resources and effort need to be invested in setting up a new BIMS every time a new study is carried out or a new biobank is being established. This duplication of effort could be avoided if a generic approach could be found that could easily be customised. In addition, the increased drive to incorporate research data from experiments into the biobanks poses the problem of integrating information of different types and formats within the biobank itself [14]. A generic approach to storing these different types of information could ease this problem.

2.3.3 Collaboration between biobanks

There is an increased need for biobanks to collaborate and share samples and information, especially for samples of rare diseases to achieve statistical significance in the experiments [11, 104, 110, 100, 13, 111, 112, 105, 113, 101, 10, 114]. Indeed, the lack of biospecimens limits the potential for translational research [112, 105, 115], such that the pace of scientific advances cannot be matched with its exploitation in medical research [15]. In addition some fields of omic research, such as large-scale genome-

wide association studies in population genomics and pharmacogenomics require very large collections of biological samples [116], making collaboration between biobanks paramount to achieving high enough numbers of biological sample material. Another incentive for sharing of information lies in the fact that there is often an information gain associated with combining sets of data, i.e. the information contained in an integrated database is higher than when each set of data is viewed separately [14]. There are several examples of collaborating biobanks, such as the EUELC project, which is a cooperation between 12 European countries for lung cancer [117], or the GenomEUtwin European Biobank cooperation [118] between 6 European countries. Collaboration between biobanks needs to achieve harmonisation on several levels, including on the legal and ethical level [101, 119], on the level of sample handling [120, 105, 110], and on the level of information, such as integration of genotype and phenotype data collected over long time in different countries in different formats [102, 120, 16, 121, 100, 109]. A large part of research in biobanking is concerned with the development of best practice standard operating procedures for physical sample handling, such as collection, processing, preservation and storage of the samples to enable the sharing of samples between biobanks [122, 123, 120, 124, 8, 106, 114, 5].

Just like the biological material, the data that needs to be shared between collaborating biobanks has to be comparable. Thus the meanings have to be preserved and similar data controls and coding practices need to be used, otherwise the validity of research results may be affected [100]. When data is annotated poorly or incorrectly, research is inhibited and may be delayed or even result in wrong research outcomes [100]. Often, data recording standards used within biobanks combine existing local and international standards that generally only fulfil local requirements Riegman:2008p425. There is a major annotation problem in international networking of biobanks [110]. As with many existing EHR implementations, the variety of BIMS solutions makes the sharing of data between collaborating biobanks difficult [10, 100]. The insufficient effort and isolated nature of data standardisation efforts for biobanks further limits the amount and scale of scientific research that can be carried out [16, 100]. A further problem posed in international collaboration is the use of different natural languages and different interpretations of international coding systems used in diagnosis, such as ICD-10 or SNOMED [100]. The outcome of national biobanking initiatives has been a wide variety of commercial and non-commercial biobank information management solutions that vary in sophistication and functionality provided [100]. Full biobanking software solutions provide functionality for clinical annotation, samples management, sample ordering and shipping and study specific functionality [100]. However, adapting any solution for a local biobank can be difficult due to internal data problems and data problems make collaboration between biobanks is even more challenging [100].

The key to data integration is seen in the standardisation of data [16]. Currently, achieving standardised data sets in collaborating biobanks often require extensive data harmonisation efforts that are both time consuming and expensive to perform before any data mining can be safely performed, as shown in TwinNET, a federated twin database [16]. However, data and pathology reporting standards for biobanking are still only emerging (e.g. [125, 6, 16, 126]) and achieving syntactic and semantic interoperability between biobanks remains a problem [16]. For example, approaches to deal with information harmonisation and to ensure cross-institutional semantic and syntactic interoperability include the development and use of Common Data Elements (CDEs) together with a controlled vocabulary, as has been carried out for mesothelioma [7], prostate cancer [127] and pancreatic cancer tissue banking Voidonikolas:2009p441, and multidisciplinary initiatives to develop local pathology reporting standards [125]. These CDEs cover demographic patient data, epidemiological data, patient clinical history, pathology data and follow-up data such as treatment, disease recurrence and vital status of the patient [6], [7]. However, the data handling issues in biobanks are still very much an isolated and unexplored area of informatics [16].

2.3.4 Summary of Biobanking

Recent advances in biotechnology have increased the demand for high-quality, well-annotated human biospecimens, resulting in an increase in biobanking activity and requiring existing biobanks to collaborate. The information in biobanks is managed in a biobank information management system. Current biobank information management systems are based on proprietary solutions, requiring duplication of effort every time a new biobank is set up and prohibiting the sharing of information in existing biobanks.

In addition, this literature review has highlighted the importance of collaboration and data sharing between biobanks for research. However, efforts regarding data sharing between biobanks are generally localised and of an isolated nature, and are generally concerned with data standardisation, such as the definition of common data elements. With more and more biobanks being established, data sharing between collaborating biobanks will become more of an issue. Instead of isolated efforts, a generic approach to sharing information between collaborating biobanks and making this information available for future research is needed. A generic approach to data sharing between biobanks would support biomedical knowledge discovery by enabling data analysis over large and longitudinal sets of data. The electronic biomedical research record proposed in this research intends to integrate clinical patient with research data from experiments on patient samples in a generic and semantically interoperable way. Data standardisation efforts identified during the literature review into this emerging field of biobanking concentrate on the patient's clinical background and phenotypical data

and pathological reports and sample description prior to biomedical experimentation, and thus partially overlap with EHR standardisation efforts. However, no candidate standards were identified that could provide useful input for the development of the envisaged eBMRR regarding the sharing of research data won from biomedical experiments carried out on patient samples, which is the focus of this research.

2.4 Sharing omic data

2.4.1 Introduction

As previously mentioned, biomedical knowledge discovery requires the merging of clinical patient information found in medical records with results from molecular research, such as the results of high-throughput omic analyses on patient samples.

The recent emergence of high-throughput technologies highly increased the speed of biomedical knowledge discovery and is revolutionising the way in which biomedical research is carried out. With the use of high-throughput technologies it is now possible to investigate all of the proteins or metabolites present in a biosample in one experiment on a genome wide level. The consequence of using these high-throughput technologies is that they generally create high volumes of complex data, such as genomic, proteomic and other omic data, depending on which type of biomolecule is being investigated [128]. Furthermore, context or meta-data about a sample and experimental procedures is of utmost importance in functional genomics investigations due to the sensitivity of the result to the experimental technique used [129, 130].

The use of these new high-throughput technologies, such as microarrays and mass spectrometry, presents challenges to the system of scientific publication, since these experiments produce large quantities of data [131, 132, 133, 134]. The data and meta-data supporting the publication needs to be made available for readers to be able to independently evaluate the interpretations made in the publication or to reproduce the experiment or analysis [135, 136], but supporting data for omic experiments, such as raw data from microarray experiments cannot easily be presented [133]. Many journals now require that data supporting publications is made freely and publicly available, for instance by depositing it in suitable public databases [136, 137].

The increasingly common usage of high-throughput technologies in research laboratories, together with the need for researchers to share their data and results inhibited by the existence of incompatible and/or proprietary data formats, have lead to the creation of omic data reporting standards. These standards support the sharing of omic data between investigators, promote best practice by allowing for independent evaluation of the study and reinterpretation of the results, and promote the development of

new data analysis algorithms [138, 139] and reuse of data for further research, helping to generate new knowledge (e.g. [140, 141]). A further benefit of these standardisation efforts is that they enable large-scale omic data mining.

While there are ongoing development efforts for *openEHR* archetypes to support the recording of clinical information, there are no known efforts for developing *openEHR* archetypes to support the recording of results from molecular research on patient samples. A fundamental part of advancing biological and biomedical research is that information about an experiment is recorded and reported in a structured and unambiguous way so that it can be shared, integrated and reused more easily [142, 143]. To support the development of *openEHR* archetypes for molecular research, the content and structure of the information to be recorded needs to be investigated.

The remainder of this chapter provides a background to and overview of existing omic data reporting standards. In the following sections, after providing a short history, a detailed description of omic recording standards is presented, broadly divided by their domains, including the organisations behind their creation and maintenance, their implementation and adoption. Finally, an overview of initiatives for cross-omic standards development and harmonisation is presented.

2.4.2 Short history of omic data reporting standards

The history of omic data standards closely follows the history of omic technologies. The success of the gene sequencing technologies in the field of genomics led to the emergence of other omic techniques, which are focussing on investigating gene function. Arguably the first omic technique that emerged was microarray technology. Microarrays can be used to investigate gene transcription on a genome wide scale, producing gene expression patterns for tissues or cells, leading to the field of transcriptomics. The rise in high-throughput techniques for the investigation of later gene expression products, peptides and proteins, gave rise to the field of proteomics.

The first major need for an omic data recording standard arose in the field of transcriptomics in the late 90s, when it had become the biggest source of genomic data in the life sciences due to the decreasing cost of microarray technology [144, 145],[146]. This need led to the development of the first omic data reporting standard, Minimum Information for A Microarray Experiment (MIAME) in 2001 [18]. The development of MIAME initiated similar data reporting standardisation efforts in other biological areas, such as proteomics ([147]), metabolomics ([148]), in situ hybridisation experiments ([149]) and other high-throughput biomedical technologies, such as flow cytometry ([150]) [151, 152].

Proteomic data reporting standards were slower to emerge than standards for recording microarray data in transcriptomics, mainly due to the large heterogeneity

in proteomic techniques, but this changed with the formation of the Human Proteome Organization’s Proteomics Standards Initiative (HUPO PSI) [153], leading to the specification of the “Minimum Information About a Proteomics Experiment” (MIAPE) guidelines and associated modules [20]. Similarly, the Metabolomic Society has developed reporting standards for metabol/nomics data [148] and more recently the Genomic Standards Consortium has created standards for a richer annotation of genome sequences [154].

Functional genomics or systems biology approaches investigate the function of genes with respect to phenotype, location of expression, molecular interaction and changes in expression upon external stimulation, often combining a variety of high-throughput techniques in a single study to investigate the direct and indirect gene expression products, molecular interactions with other molecules such as RNA, DNA, metabolites and proteins, and cell environment [144, 129, 138]. With the trend towards functional genomics or systems biology approaches, it became clear that data standard development cannot happen in isolation within communities of users of high-throughput technologies, but needs to be coordinated to enable the integration of data from different omic domains [155, 156]. With this realisation, several initiatives were launched to coordinate data standards development, setting up dialogue between different omic standard developing communities and organisations. These aimed to harmonise between existing omic data standards and avoid overlap [155], such as MIBBI[157], FuGE[129] and OBO[98].

The use of omic standards is also promoted by data management and sharing policies. For example, many research organisations or funding bodies, e.g. the UK Biotechnology and Biological Sciences Research Council (BBSRC)³⁵ require that datasets produced through work funded by the body are made available in standardised form using community standards³⁶.

2.4.3 Reporting standards in Transcriptomics

DNA microarray analysis is a widely used high-throughput technique to measure the locations and levels of gene expression on a genome wide rather than single-gene scale through the generation of gene expression profiles [145, 132, 131, 133]. Microarray technology was the first high-throughput technology that could investigate different cell types and different biological conditions of different individuals all in one experiment [152].

Microarray experiments are characterised by the huge volumes of data that they

³⁵BBSRC, <http://www.bbsrc.ac.uk/>

³⁶BBSRC data sharing policy, <http://www.bbsrc.ac.uk/organisation/policies/position/policy/data-sharing-policy.aspx>

produce [144]. Microarray analysis data is complex and is only meaningful if it is viewed in context, i.e. with regard to the state of the organism from which the biomaterial was received, and the experimental conditions used to generate the data and the data analysis technique used, thus requiring a large amount of metadata to accompany the data [18, 152]. In addition, to facilitate comparison, the raw data of microarray gene experiments needs to be provided to remove the possibility of variation of data through different data normalization techniques (not to be confused with database normalization) [144, 146, 137].

However, before the development of reporting standards, gene expression data from microarray analysis was widely inaccessible to the research community, it may be found scattered over the internet on the websites of authors of publications researching gene expression levels [133], in different, non-standardised formats. These may be lost in future or may not even be accessible at all [18, 144, 137]. In addition, annotation levels of microarray data were generally insufficient to allow reuse of the data, and many publications on microarray experiments did not provide supporting data and contained insufficient information about experimental procedures [144], and other important information, such as evidence on quality, reliability and possible error levels of the generated data may have been missing [18]. Furthermore, the existence of many different formats, units, different normalisations, different microarray platforms and different experimental designs, was making the integration and comparison of microarray data difficult and error-prone [158, 159, 151] and limited the development of scalable high-throughput automated approaches for data mining and analysis, severely limiting the speed at which biomedical discoveries from microarray analysis could be made [18].

This problem did not exist for genome sequence data, for which many different formats existed in databases and analysis tools [18]. While the Human Genome Project and other DNA sequencing efforts also produced large amounts of data, genome sequence data is much simpler and more robust than microarray gene expression data, since an organism only has one genome but many different transcriptomes, depending on different physiological conditions [152, 18].

MGED and the MIAME standard

The Microarray Gene Expression Data (MGED)³⁷ Society originated in 1999 as a grassroots movement by the microarray community (users, developers, vendors) [159, 144, 151], concerned with the establishment of standards to support the sharing of microarray gene expression data, including proteomics array data [151].

MGED created the MIAME standard, the earliest reporting standard developed in the omic domain, the first version of which was published in 2001 [18, 160]. MIAME

³⁷MGED, www.mged.org, now FGED, www.fged.org

is now on version 2.0 ³⁸.

MIAME was designed with two principles in mind [18]. Firstly, it should specify the minimum information that would be needed to ensure the interpretability of the results of a microarray experiment, to be able compare the results to other microarray experiments and to independently verify the results of the analysis, e.g. by replicating the experiment [18]. Secondly, MIAME was to be structured so that it would be amenable to the development of automated data analysis and mining and other useful querying [18].

A microarray experiment consists of hybridisations between samples and arrays. The resulting hybridised arrays are scanned, producing an image. This image is analysed further, resulting in a set of measurements for each array element. A normalisation is carried out on the measurements and the data sets of replicate hybridisations are combined.

Figure 2.12 shows a schematic representation of the six components of a microarray experiment.

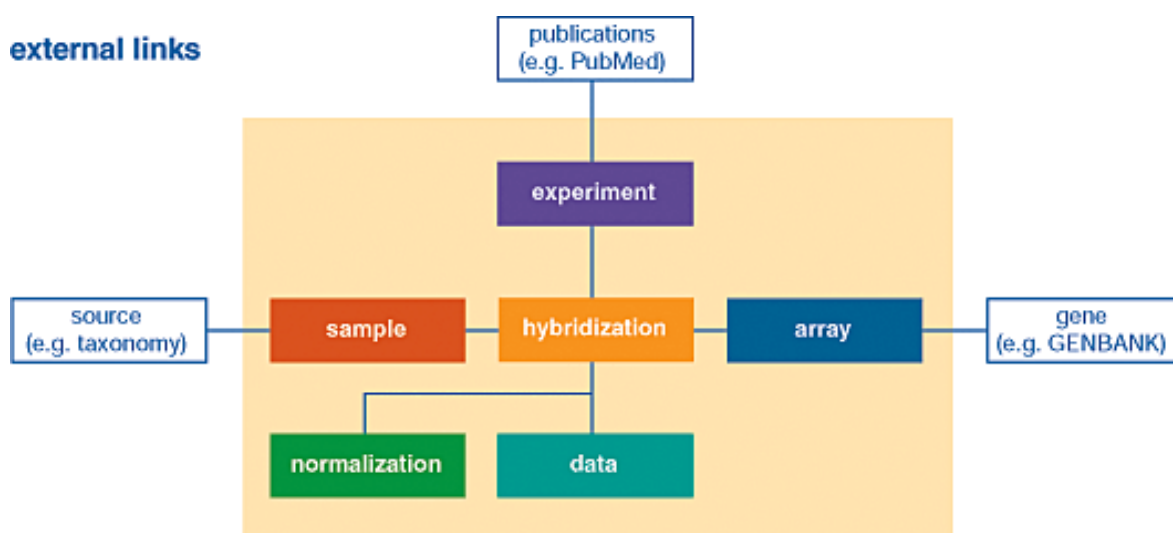


Figure 2.12: Schematic representation of the six components of a microarray experiment (Source: [18]).

These six components of a microarray experiment are used as the basis for the MIAME standard, so that MIAME is divided into the following six parts (from [18]):

1. **Experimental design:** the set of hybridisation experiments as a whole
2. **Array design:** each array used and each element (spot, feature) on the array
3. **Samples:** samples used, extract preparation and labelling

³⁸MIAME 2.0, <http://www.mged.org/Workgroups/MIAME/miame.2.0.html>

4. **Hybridizations:** procedures and parameters
5. **Measurements:** images, quantification and specifications (the raw data)
6. **Normalization controls:** types, values and specifications

MIAME only specifies the content that needs to be recorded, but not the format in which it should be recorded, i.e. it does specify how it should be implemented technically and does not impose a requirement to use a particular platform, software or method of data analysis [18]. However, together with the Life Sciences Research Task Force of the Object Management Group (OMG)³⁹, MGED also developed a data model that builds the basis for a standard data-exchange format called MAGE-ML that implements MIAME requirements to allow communication of MIAME data between laboratories, archives and analysis packages and supporting software is also developed by participating organisations [18].

In addition, the MGED Ontology Working Group have produced an ontology (the MGED ontology, MO) and rules for the unambiguous annotation of MIAME compliant microarray gene expression data to be used in conjunction with MAGE-ML [161, 162, 151, 18]. The MO covers all areas of microarray experiments, including experimental design, array layout, sample preparation and hybridisation protocols and data analysis [161, 162]. MGED also created a short set of guidelines and a checklist that authors, reviewers and journal editors could use to achieve and check for MIAME compliance [163, 136, 160, 164].

Three extensions to MIAME have been developed so far [155]:

- MIAME/Tox [165] was created by the MGED Toxicogenomics working group, to specifically capture toxicology data in an experiment (e.g. histopathology, clinical chemistry)
- MIAME/Env [166] has been developed to capture the data of environmentally relevant organisms in functional genomics investigations
- MIAME/Nut [167] has been created to capture nutritional components of functional genomics investigations

MIAME data formats

To facilitate the standardised storage and exchange of MIAME compliant microarray gene expression data, MGED have developed the several technical artefacts: MAGE-OM (microarray gene expression object model), a UML based object model repre-

³⁹OMG, <http://www.omg.org>

senting MIAME concepts and their relationships, MAGE-ML, an XML-based data exchange format derived from the MAGE-OM to support exchange of MIAME compliant gene expression data and which has subsequently been implemented in microarray data capturing software and a Perl, Java and C++ based software tool kit (MAGE-STK) to provide APIs to facilitate the creation of MAGE-ML documents [19].

While MAGE was adopted reasonably successfully in microarray repositories at first, the MAGE-OM was considered too complex and use of MAGE-ML too difficult in laboratories without bioinformatics support [152, 168], and too cumbersome due to the increased size of data files based on XML [151, 168]. MAGE-ML has also been criticised for being too flexible, allowing the same information to be described in different, inconsistent ways [128, 135, 151]. Also, MAGE does not in itself guarantee compliance with MIAME as valid MAGE documents can be created that contain less or more of the information required by MIAME, depending on the individual's interpretation of MIAME [135]. To solve these problems, MGED later developed MAGE-TAB (MicroArray and Gene Expression TABular), a simple spreadsheet-based, tab-delimited format to report MIAME-compliant data [168]. The MAGE-TAB format became more popular, as it could be edited directly without the use of special tools and because of software support for data import, e.g. for import into the Bioconductor analysis tool [152]. However, a 2009 review of MIAME showed the MAGE-TAB adoption was low in comparison adoption of the MIAME checklist [152]. MGED later revised MAGE to eliminate existing ambiguities [151] and in 2006, MAGE-ML v2⁴⁰ was published, based on FuGE (see Section on Standard harmonisation). However, no single implementation of MAGEv2 is known and MAGE-TAB is the universally used format, having completely replaced the more complex MAGE-ML (v1)⁴¹.

MIAME adoption in journals and databases

Adoption of the MIAME guidelines for journal publications received a boost after the MGED society wrote an open letter in 2002 to major journals, such as Nature and the Nature research journals, Lancet, Science and Bioinformatics, to require that authors of microarray papers provide MIAME compliant metadata either as a database submission, e.g. to ArrayExpress or GEO or as supplementary data as a prerequisite for publication. The journals have agreed to follow this recommendation [163, 136, 160, 164]. Major scientific journals now require the submission of MIAME compliant data to support publications of microarray experiments [151]. Moreover, several funding agencies (e.g. UK's Natural Environment Research Council (NERC)) require adherence to MIAME guidelines for all microarray data produced under funded programmes [151].

⁴⁰MAGE-ML v2, <http://www.mged.org/Workgroups/MAGE/magev2.html>

⁴¹personal communication with Ugis Sarkans from FGED, ugis@ebi.ac.uk

MIAME has been implemented in major microarray databases, e.g. ArrayExpress, GEO and the Center for Information Biology gene EXpression (CIBEX) database⁴² [169]. ArrayExpress supports submissions in MAGE-ML format and via MIAMExpress, an online submission tool aimed mainly at users who cannot provide submissions in MAGE-ML format [144, 170]. The ArrayExpress relational database schema is modelled on the MAGE-OM [171, 172, 173]. ArrayExpress also supports MAGE-TAB submissions [152]. MIAME is a set of guidelines only and thus the level of detail required to satisfy MIAME is subjective and may be interpreted differently. For example, even if all mandatory fields are provided, there is no way to check if the data is accurate and sufficient and thus it is not possible to automatically enforce the guidelines and validate submissions for compliance [174, 175]. Both ArrayExpress and GEO provide manual validation of MIAME compliance [175, 174, 156]. The Stanford Microarray Database provides software allowing for the upload of MAGE-ML formatted documents of companies producing microarray chips and facilitating the submission of MIAME-compliant, MAGE-ML-formatted data to the GEO and ArrayExpress databases [176].

Other transcriptomic standards

Together with decreasing costs, new-generation sequencing (NGS) techniques become a viable alternative to traditional microarray applications, such as gene expression profiling and gene copy number and epigenetic variation analyses [152]. The MGED Society has responded with the development of new guidelines for Minimum Information about a high-throughput SeQuencing Experiment (MINSEQE)⁴³, which are already being supported by GEO and ArrayExpress [152]. The Minimum information specification for in situ hybridisation and immunohistochemistry experiments (MISFISHIE) provides guidelines for the reporting of experiments that investigate the localisation of gene expression in tissues through visual interpretation, e.g. IHC, ISH, lectin affinity histochemistry and experiments using reporter gene constructs, such as green fluorescent protein (GFP) and β -galactosidase [149]. MISFISHIE was inspired by MIAME, and can be used in conjunction with other MI specifications, such as MIAPE and MIAME, to support integrative functional genomics [149]. Figure 2.13 shows a schematic representation of the six parts of MISFISHIE. The authors of MISFISHIE also developed a MISFISHIE-compliant XML-based data exchange format and a document type definition (DTD) for several gene expression repositories [149].

⁴²CIBEX, <http://cibex.nig.ac.jp/>

⁴³MINSEQE, <http://www.mged.org/minseqe>

MISFISHIE

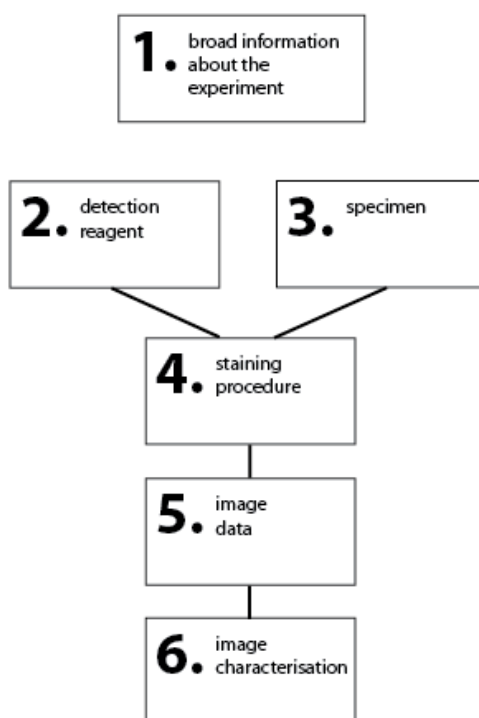


Figure 2.13: Schematic representation of the six components of MISFISHIE (Source: <http://www.emouseatlas.org/emage/info/misfishie.html>)

2.4.4 Reporting standards in Proteomics

Proteomics comprises a family of techniques used to investigate the proteins and peptides in a biological sample [147]. Proteomics data is characterised by large volume and complexity [143], just like microarray data in the domain of genomics. Like genomics, transcriptomics and structural proteomics, the field of proteomics is organisationally evolving from small-scale local research towards a discipline characterised by large-scale, highly automated investigations (such as described by Washburn et al. [177]) and large-scale collaborations, such as the Human Proteome Organization (HUPO) Proteome Projects [178, 179, 180]. In the last decade, significant advances have been made in the field of proteomics and the generation and analysis of proteomics data is widespread with high-throughput proteomic technologies now commonly used in scientific research [180, 181, 20, 147].

Experimental metadata plays an important role in proteomics experiments [20]: different cells express different genes under different conditions, thus there are many different subsets of a proteome [20], similar to there being many different transcriptomes of the genome. Furthermore, the results of proteomic analysis depends on how the

analysis was carried out. For example, protein identification by mass spectrometry is sensitive to the particular separation techniques involved, the equipment that was used, the protein identification tool and the particular database that was queried to assign protein identities [20]. Thus, to be able to make sensible comparisons between datasets or to carry out valid statistical analyses on aggregated data, metadata (the data about the data), i.e. the biological context, such as sample provenance and experimental conditions on how analyses were performed, need to be provided to avoid unsuitable experimental design and the occurrence of random and systematic errors [20]. Incomplete or ambiguous annotations of proteomics data sets can lead to invalid comparisons and inaccurate or misleading summary statistics [143, 147].

However, while lots of proteomics data was being produced, it was not recorded in a systematic way [180]. Many proprietary formats existed that required the original vendor's software to read in proteomics data owing to the commercial origin of many proteomic techniques [147]. Proteomics data recording was characterised by the use of incompatible or computationally unsuitable data formats (e.g. PDF files), and if it was being made available at all, its storage was largely fragmented with data being scattered on the internet on the websites of researchers and journals or local databases [153, 180]. The lack of systematic capture of proteomics data in public databases meant that validation of results published in the scientific literature was made difficult and data could not be easily reused for further research [153].

Standard development in the proteomic domain was inhibited by the large complexity, diversity and fast evolution of techniques in the proteomics domain [153]. While the need for a systematic way to record and report proteomics data was recognised by the scientific community [153] and progress was made on an individual basis towards the development of reporting guidelines and data formats for proteomics data [182, 183, 184, 185, 186], it was not until the establishment of the HUman Proteome Organization (HUPO) Proteomics Standards Initiative (PSI) ([187, 153]), that the development of proteomic reporting standards began truly to move forward.

The HUPO PSI

The main standardisation work in the proteomics domain is being carried out by the HUPO through the PSI working group [188]. HUPO⁴⁴ ([189]), a community of industry, academia and government groups, was founded in 2001 to merge existing national and regional proteome organisations and to support large-scale, public proteome projects [143, 147]. The Proteomics Standards Initiative (PSI)⁴⁵ was founded as a working group of the Human Proteome Organisation (HUPO) in April 2002 with the goal to create

⁴⁴HUPO, www.hupo.org

⁴⁵PSI, <http://www.psdev.info/>

community standards for the representation of proteomics data, to enable systematic recording of proteomics data, proteomics data exchange, comparison and validation [153, 187]. The PSI is an open community initiative and volunteer participants come from both the academic and commercial area, including experimentalists and bioinformaticians, instrument vendors, data producers, funding organisations and publishers, interested in the development of proteomics standards [153]. The PSI developed the “minimum information about a proteomics experiment” (MIAPE [143, 147]) guidelines, specifying the minimum data and metadata that should be recorded about a proteomics experiment [187], which were published in 2007 [20]. While MIAME was the first major reporting standard in the genomic domain, MIAPE [20] was the equivalent first major reporting standard in the proteomic domain.

MIAPE

The MIAPE guidelines consist of a parent document and a number of modules, each covering a specific technology, and each coming with a checklist [20].

MIAPE consists of a stable ‘parent’ MIAPE document that describes the general guidelines on the information that needs to be provided about the use of a technique and the data that was generated from it, and several more labile modules, each relating to a particular (group of) technology [143, 20, 147]. Modules are newly created as new experimental techniques appear or existing ones changed in response to changes in the experimental technique(s) [20].

Module development by HUPO-PSI is harmonised, having resulted in the creation of a number of compatible data exchange formats that are widely used and supported in software for data processing and in public repositories [190].

The PSI has so far published seven MIAPE modules, describing minimum reporting requirements for experiments using:

- Capillary Electrophoresis (MIAPE-CE) [191]
- Column Chromatography (MIAPE-CC) [192]
- Gel Electrophoresis (MIAPE-GE) [190]
- Gel Image Informatics (MIAPE-GI) [193]
- Mass Spectrometry (MIAPE-MS) [194]
- Mass Spectrometry Informatics (MIAPE-MSI) [195]
- Molecular Interactions (MIMIx) [196]

For example, MIAPE-CC covers the following six areas of a column chromatography experiment (adapted from [192]):

1. **General features:** analyst details, description of the sample, sample preparation and the injection procedure.
2. **Description of the column(s) used:** product details and physical characteristics including the stationary phase, and the chromatography system used for the separation.
3. **Mobile phase:** the concentrations of each of the mobile phase constituents.
4. **Properties of the column run:** time, gradient (with reference to the mobile phases), flow rate and temperature.
5. **Pre- and post-run processes:** equilibration, calibration or washing.
6. **Column outputs:** chromatogram; details of fractions collected.

MIAPE has been developed by balancing sufficiency and practicability: enough information needs to be supplied about the data and the experimental conditions so that an experiment and its interpretation and conclusions drawn can be understood and evaluated and repeated by an independent party, but at the same time MIAPE compliance should be practical to achieve without being overly work intensive to the producer as to inhibit its widespread acceptance [20].

MIAPE has been modelled on MIAME: MIAPE provides a specification for an implementation-independent description of information [20]. MIAPE does not impose a specific file format to achieve compliance [20]. Thus, in addition to the MIAPE specifications, the PSI also develops accompanying XML-based data exchange formats (e.g. mzData for MS) to support the electronic exchange of proteomics data, which can be used to implement minimum information requirements [153, 143, 147] and controlled vocabularies or ontologies [153].

MIAPE Data exchange formats

PSI have developed a modified Microsoft Excel spreadsheet for each module as a base solution for low volume data submissions, similar to the ProteomeHarvest tool for submissions to the PRIDE database [134, 20]. Further, the PSI develops stable XML schemas, with new versions to be released about every two years to allow implementation in software, and dynamic controlled vocabularies (CVs) that can be updated continuously, thus balancing stability versus dynamic development [153]. Some of these are briefly introduced in the following paragraphs.

The first successful PSI developed data exchange format was the Molecular Interaction Format (MIF) [197]), developed by the Molecular interaction workgroup. This format later became the stand-alone MIMIx specification [147]. It solves the problem of the ambiguous use of identifiers for molecules in recording interaction experiments, e.g. gene names [196]. The PSI MI standard was developed in collaboration with major public and private sector organisations and MI data in PSI MI XML format has been implemented in major repositories and tool support, such as Cytoscape⁴⁶ [153, 198, 199]. Furthermore, HUPO PSI also developed MITAB2.5, a simpler, tab-delimited format that can be used in conjunction with Microsoft Excel [199]. As part of PSI-MI, a comprehensive controlled vocabulary was defined to standardise the annotation of data objects [199, 196].

Another major success in data formats by the PSI is the mzData format developed by the Mass Spectrometry Standards Working Group, capturing the description of a Mass spectrometer, its parameters and the (processed) data generated (from “raw” profile data to heavily processed peak lists) [147, 200]. Close collaboration with instrument vendors lead to mzData to become the main format for sharing MS data in proteomics community for recording mass spectrometry results, being implemented widely in mass spectrometry instruments, search engines and databases [147, 153]. mzData was later merged with the mzXML format ([183]) to become mzML, to be used with the PSI-MS ontology [147]. PSI have further developed AnalysisXML for the representation of MS search results and extensions to this format covering the analysis of mass spectra for peptide and protein and other metabolite identification and quantitation [147]. PSI is also developing a data exchange format for mass spectrometry based proteome quantification (mzQuantML) which may be used as basis for capturing data for gel image informatics [201]. Standard development is ongoing in PSI, current efforts include e.g. mzQuantML and standards for protein identification, such as the mzIdentML data exchange format for proteins and peptides identified from mass spectra [202].

The PSI Gel working group is responsible for the development of reporting requirements, controlled vocabularies and data formats for the recording of gel electrophoresis experiments and the informatics analysis performed on resulting gel images [203, 201]. So far, it has developed the MIAPE GE, MIAPE GI specifications, the GelML data format and the Sample Processing and Separation techniques controlled vocabulary (sepCV) [201]. GelML covers the description of electrophoresis protocols, direct and indirect (e.g. through Western blotting) protein detection and acquisition of the image from the gel matrices [201]. The GelML format is based on MIAPE-GE and can be used to create MIAPE-compliant documents [201]. GelML is an extension to the FuGE data model, allowing the easy adaption of existing FuGE software for GelML [201].

⁴⁶Cytoscape, www.cytoscape.org

MIAPE adoption by journals and databases

Shortly after publication of the MIAPE standard in 2007, major journals, such as Nature Biotechnology recommended the use of MIAPE guidelines in publishing proteomics data and the deposition of raw data from molecular interaction and proteomics experiments in a public repository as part of submitting the manuscript [204, 143]. mzData has been adopted by device vendors, software developers and data repositories, such as the European Bioinformatics Institute’s Proteomics Identifications Database (PRIDE) [180]. PRIDE accepts mass spectra submissions in PRIDE XML format or mzData format and intends to support analysisXML upon release [134]. PRIDE complies with PSI and uses ontologies and controlled vocabularies for submission annotations [134]. Public proteomics databases, e.g. as PRIDE, Gelbank and Swiss2DPage are planning to support MIAPE-GE [190]. The ChEMBL repository⁴⁷ has mapped much of its content to the standards and formats of HUPO PSI-MI [205, 198]. Reactome use a new binary data exchange format, based on PSI-MITAB [199].

Other proteomics guidelines/standards

Additional proteomic guidelines and data formats are developed by other proteomic community initiatives. For example, the protein-affinity community developed MIA-PAR, specifying the minimum information that needs to be recorded about a protein affinity reagent [206]. MIABE (Minimum information about a bioactive entity) specifies the minimum of information that should be reported for bioactive entities, such as drugs, pesticides and food additives to support the development of improved molecules, e.g. through computational analysis of molecules to create more successful drugs [198]. The developers of MIABE recommend to use an extended version of the existing HUPO-PSI MI (Molecular Interaction) XML 2.5 interchange format as data exchange format and use an extended version of the accompanying HUPO PSI-MI CV to fully describe properties of a bioactive molecule as primary source for annotating bioactive entity data [198]. Another community developed proteomic standard is MI-FlowCyt, specifying the minimum information that should be reported about Flow Cytometry (FCM) Experiments [207]. MIFlowCyt was adopted by the International Society for Advancement of Cytometry (ISAC) has adopted MIFlowCyt [207]. MI-FlowCyt covers specimen description, reagents used in the FCM experiment, FCM instrument configurations, and data processing methods of the raw data [207].

⁴⁷ChEMBL repository, <https://www.ebi.ac.uk/chembl>

2.4.5 Reporting standards in Genomics

With technological advances towards ever faster and cheaper sequencing technologies, especially new high-throughput technologies, new genomic data is produced exponentially [208, 209]. With it, an increased need for a new data standard for capturing a richer set of context-dependent meta-information on complete genome sequences, such as ecological environment and evolutionary relationships lead to the recent development of the MIxS group of standards for genomic data [22], consisting of MIGS [210] and MIMS [211] and MIMARKS [22].

The MIxS group of standards

The MIxS group of standards was developed by the Genomic Standards Consortium (GSC) [22]. The GSC is an open-membership organisation that promotes community-based standardisation efforts in the genomic domain for the annotation of genomes to enable the integration and sharing of genome sequence data [154, 212].

The Minimum Information about a Genome Sequence (MIGS) standard specifies the following seven types of information to be recorded (adapted from [210]):

- Genomic Features
- Taxonomy
- Ecology
- The annotation process
- Origin and availability of strain
- Related publications and electronic resources
- Contact information

MIMS (Minimum Information about a metagenome sequence) is an extension of MIGS for the annotation of metagenomes [211]. The MIMARKS (Minimum information about a marker gene sequence) specification is an extension to MIGS and MIMS and unifies the three MI checklists under the MIxS (minimum information about any (x) sequence) standard [22]. The MIxS standard captures core contextual information such as geolocation and habitat and information on sequencing technique used [212]. Figure 2.14 provides a schematic overview of the MIxS standard.

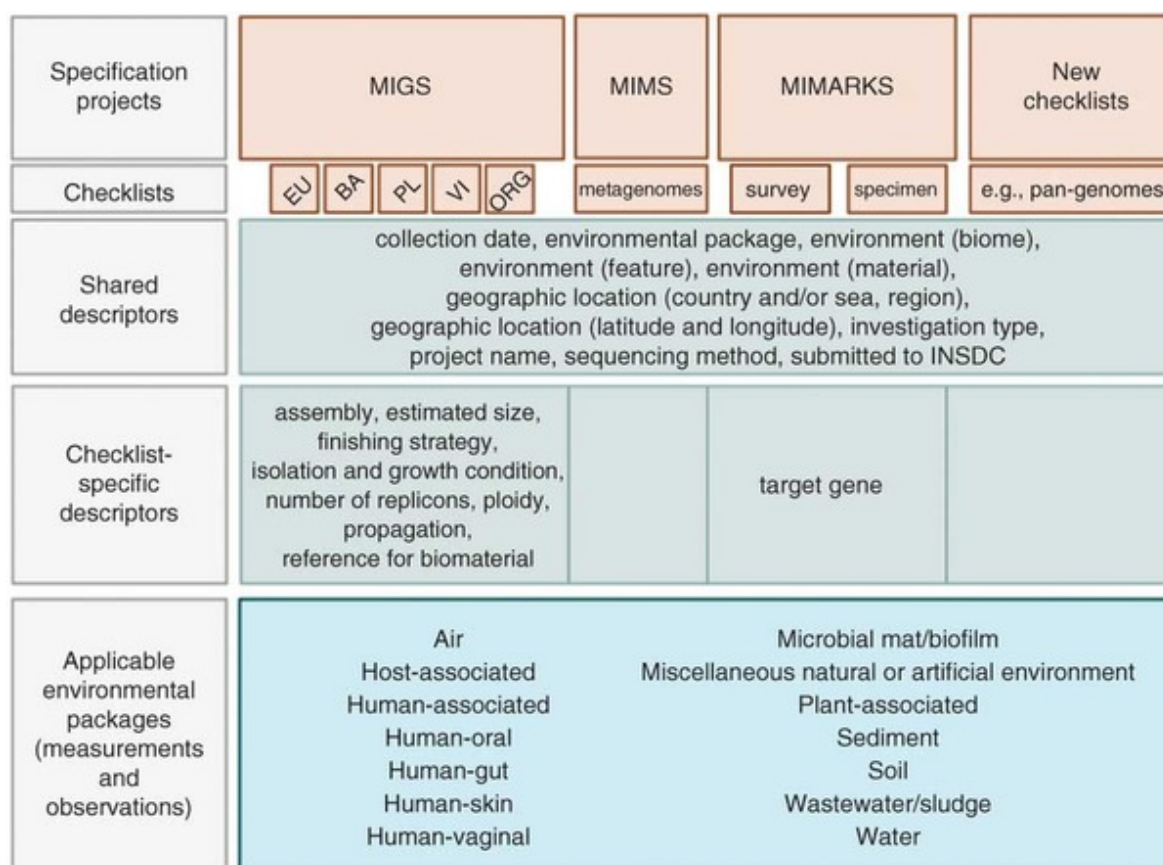


Figure 2.14: Schematic overview about the MIxS standard (brown), including combination with specific environmental packages (blue). EU, eukarya; BA, bacteria/archaea; PL, plasmid; VI, virus; ORG, organelle. (Source: [22]).

In 2008, the GSC developed the XML-based “Genomic Contextual Data Markup Language” (GCDML) which now implements all MIxS specifications and can be used to create compliant reports for data exchange and storage [23, 213, 22]. GCDML is sample-centric, and allows for information about provenance and the processing of a biospecimen, from collection to sequencing and analysis to be captured [23]. MIxS has been adopted and is supported by major sequence databases and informatics resources [22], for example by INSDC [212, 22], GenBank and the EBI-ENA (EBI European Nucleotide Archive) [22]. The GOLD database has developed GOLD-specific controlled vocabularies to represent metadata to enable compliance with MIxS, [214].

MIGen

MIGen, the Minimum Information about a Genotyping Experiment specification⁴⁸, was created by inter-disciplinary experts in basic and clinical biological research, biostatistics and bioinformatics [215]. Genotyping is performed widely both in clinical and

⁴⁸MIGen, <http://migen.sourceforge.net>

basic research, and encompasses a huge variety of different techniques and methodologies for genotyping assays, ranging from simple PCR for single genomic regions to high-throughput technologies assessing genome-wide variations, leading to high volumes of complex data [215]. The MIGen standard includes reporting requirements about the design of the experiment, the description of the subject, the specific genotyping technique used, data analysis and quality control of the result. Due to the large diversity in genotyping experiments, MIGen is organised hierarchically into top-level components covering general information that applies to all genotyping experiments and modules for specific experimental techniques or analysis methods [215], as shown in Figure 2.15. There is some partial overlap between MIxS and MIGen, both using a different checklist architecture, e.g. with regard to sample description [215].

MIGen Package

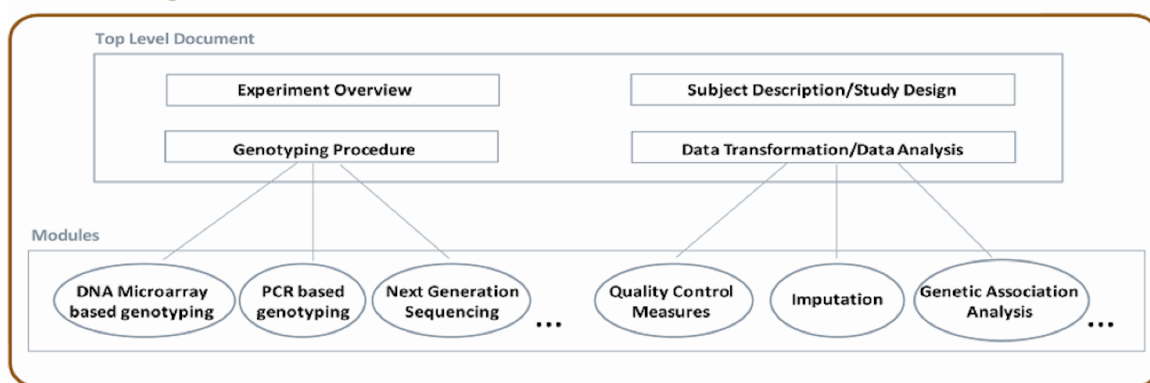


Figure 2.15: Hierarchical organisation of MIGen (Source: [215]).

2.4.6 Standard development in Metabolomics

The field of metabolomics is concerned with investigation of metabolic changes on an omic scale to elucidate biochemical reactions and biological functions of systems of cells [148]. The results of metabolomics investigations (consisting of the biological perturbation of a system and the measuring of the elicited metabolic responses) critically depend on the state of the system's biology and how the experimental measurements are being performed and recorded [148]. The Metabolomics Society⁴⁹ has started an initiative to standardise the reporting of metabol/nomics data to harmonise the reporting between the related fields of metabolomics, metabonomics, “metabolite profiling,” “metabolic fingerprinting,” “lipidomics,” etc. [148]. Due to some overlap with the domains of proteomics (such as the use of mass spectrometry) and transcriptomics in the areas of data transformations and processing existing data formats can be shared [148].

⁴⁹Metabolomics Society, <http://msi-workgroups.sourceforge.net>

2.4.7 Collaboration, coordination and harmonisation in the development of omic standards

Since the development of MIAME, many groups from different omic disciplines have started to develop omic data reporting standards [128], often modelled on MIAME, such as reporting standards for metabolomics [148], proteomics [147], in situ hybridisation, immunohistochemistry [216], flow cytometry [150], and experiments centred around a specific technology [155].

Until recently, the development of these standards has happened in an uncoordinated, isolated fashion, leading to duplication of efforts resulting in the proliferation of MI specifications, associated data exchange formats and ontologies that are incompatible and overlapping, making re-use of data and integration of data for analysis difficult because of differences in scope, structure and vocabulary [135, 156, 217, 157, 155, 129, 128].

In fact, many parts of a scientific investigation are shared by many areas, such as sample collection and study design. This has created overlaps between standards. For example, the scope of MIAPE overlaps partially or completely with minimum reporting requirements of other scientific domains, such as those in transcriptomics [18] and metabolomics [148] in study design, collection procedures for biosamples, statistical analysis used and in the use of specific analytical technologies [143]. Furthermore, many techniques are being used in other scientific domains, for example, both proteomics and metabolomics use chromatography and mass spectrometry [20, 147, 143]. The Core Information for Metabolomics Reporting (CIMR), MIAME and MIGS specifications overlap in the description of the biosample, and others overlap in specific technologies, e.g. mass spectrometry is covered both by CIMR and MIAPE [156].

With the trend towards multi-omics approaches that may use a number of different omic approaches and technologies, such as proteomic, transcriptomic and metabolomic, to investigate the same source sample, the integration of the different types of omic data generated by these approaches is becoming more and more important, to enable effective data analysis [156, 135, 217, 138, 147, 129, 130, 155]. For example, nutrigenomics, environmental genomics, and toxicogenomics investigations cannot be described sufficiently by any individual of existing reporting structures [155]. Furthermore, integration of data is a prerequisite for enabling its reuse [157]. The number of data- rather than hypothesis-driven studies has been growing over recent years, making coordination and harmonisation between groups involved in the developments of data standards and exchange formats more and more important [218, 157]. Standard development should be coordinated to avoid the creation of multiple, competing standards and to avoid overlaps, but also to avoid incompatibility where there is no overlap in

scope, e.g. with regard to the structure, granularity of information recorded, terminology etc. which can hinder data integration and the combination of diverse sets of data [156, 157, 155, 148, 149].

There have been efforts by individual standards development organisations to avoid overlap. These efforts include coordinating the development of reporting requirements for technologies, protocols and entities that are common in a lot of areas in collaboration between standard bodies, by creating shared modules, as for MIAPE [143, 20]. In the development of MIAPE, the PSI concentrated on techniques specific to proteomics rather than very general areas (such as the description of the biosample used in the study) in order to avoid overlap with other standards such as MIAME to facilitate integration of different data sets [20]. Furthermore, there are efforts of individual standards development organisations to collaborate and coordinate with other organisations in the development of standards, such as by the PSI [147], the Genomics Standards Consortium [130], and the Metabolomics Society [148, 147, 219, 148].

More recently, several overarching collaboration and coordination efforts to harmonise amongst standards have been initiated on all levels of standard: on the level of MI specifications (MIBBI), on the level of data exchange formats, representing syntax (FuGE, ISA-TAB) and on ontologies, representing semantics (OBI, OBO Foundry) [156, 157]. Each of these efforts is open to the public, collaborates with diverse biomedical domains, and participants of each effort interact to foster coordination between these efforts [156]. These efforts are described below.

Coordination at the level of the specification

In the following sections, the initiatives MIBBI, the RSBI, the GSC, and their efforts in coordination among MI specifications are described.

MIBBI: Coordination for MI Specifications Due to the proliferation of minimum information checklists, the Proteomics Standards Initiative (PSI), the Reporting Standards for Biological Investigations (RSBI) and the GSC [157] have initiated the Minimum Information for Biological and Biomedical Investigations (MIBBI)⁵⁰ project [157, 143, 20]. MIBBI aims to coordinate the creation and harmonisation of minimum information guidelines and checklists between the different biosciences domains, to avoid the duplication and overlap of existing standards [157]. Over 30 projects developing reporting requirements are registered under MIBBI. Figure 2.16 illustrates the current overlap between checklists.

⁵⁰MIBBI, <http://mibbi.org/>

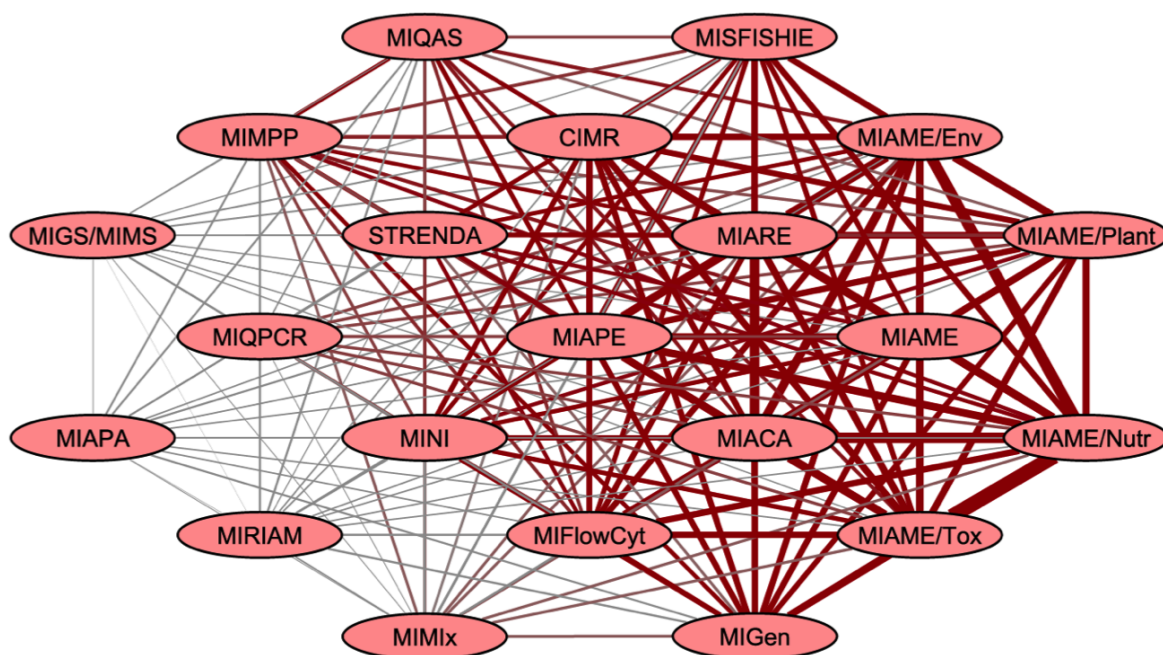


Figure 2.16: Interaction graph for MIBBI projects (line thickness and colour saturation show similarity) (Source:http://www.docstoc.com/docs/document-preview.aspx?doc_id=118652775.)

MIBBI has two key parts [157]:

- The MIBBI Portal: MIBBI provides a freely accessible web-based resource acting as a portal or ‘one-stop shop’ for exploring existing minimum checklist projects, checklists and their associated data formats, ontologies and controlled vocabularies, databases and tools [157].
- The MIBBI Foundry: an initiative to foster the joint harmonisation of checklists, with the aim to ultimately develop a number of checklists that are integratable, clearly bounded, self-consistent and orthogonal [157]. The MIBBI Foundry is the MI equivalent of the OBO Foundry [157].

MGED: RSBI Working Group In collaboration with the PSI, the MGED Society has initiated the Reporting Structures for Biological Investigations (RSBI) working group⁵¹ in 2004 to develop standards and standard compliant software and databases [155, 147, 151, 147, 155, 151]. The RSBI working group has been formed to unite and harmonise amongst independent standard development efforts of the nutrigenomics, environmental genomics and toxicogenomics communities (such as exemplified by the MIAME extensions MIAME/Tox [165], MIAME/Env [166] and MIAME/Nut [167])

⁵¹RSBI, www.mged.org/Workgroups/rsbi/rsbi.html

to avoid duplication and incompatibility in order to facilitate data integration in functional genomics studies that use a number of omic technologies [155, 151, 135, 155, 151]. The most important output by the RSBI has been the definition of “high-level concepts” for a reporting standard that are applicable to any technical or biological application domain and which can be extended by the communities into their respective domains [155]. The use of these concepts avoids duplication of core information that is common to different biological and technical domains, including study design, sampling, sample processing and instruments [155]. The high-level concepts defined by the RSBI are as follows (adapted from [155]):

- An **Investigation** is defined as a self-contained entity of scientific enquiry, containing a holistic objective or hypothesis
- A **Design** is defined by the relationships between one or more Study(s) and Assay(s).
- A **Study** is defined as an experiment comprised of specific Phase(s), where Action(s), are applied to Subject(s) or Group(s) of Subject(s), thus contains the description of experimental procedures performed on subjects
- A **Subject** is defined as the biological sample that is being studied (e.g., a tissue slice, a population, an individual, cells).
- An **Assay** describes the experiment performed on the Subject(s), and thus contains the test(s) and data produced by omics or other technologies

These high-level concepts and their relationships are shown graphically in Figure 2.17.

The GSC: Effort to harmonise the sample concept Another group, which has later become the GSC, worked on harmonising the sample concept amongst omic domains to support multi-omics studies on the same sample [130]. Using ontological concepts, they argue that a sample may be described as an entity or “continuant” (“cellular component” in the Gene Ontology (GO) [220]), for example a mouse, a cell, a urine sample, a liver biopsy, a yeast culture or a plasmid, that exists and goes on to exist through time and may change or have “processes” (“biological process” in GO) applied to it [221], such as are RNA extraction, ageing, toxicological perturbation, and acclimatisation [130]. They propose that a sample description should contain the following components (from [130]):

- **Characteristics:** Characteristics are observable, assumed or estimated features of the entity, e.g. providing information divided into the following five sections as proposed by Gkoutos et al. [222] of an entity descriptor, a description of its attributes with associated values and units and the assay used for measuring the attributes.
- **Processes:** Processes can influence sample characteristics, e.g. the process “transportation” can change the location characteristic of a sample. In computational terms an entity on which any process has acted is regarded as a new entity, even though most of its attributes stayed the same.
- **Temporal component:** a temporal component, t , is necessary for continuants to exist. The application of a process leads to a change in characteristics at a new point in time, t_i . The temporal component can be captured in relative terms, e.g. in terms of duration since the start of an experiment or in absolute terms, such as GMT or UST, depending on what is more appropriate.
- **Environmental history:** Environmental history describes the processes that have acted upon the sample before it has been studied. This is the most loosely defined component.

The GSC also propose the use of the inheritance paradigm (cf. object-oriented programming) in the description of sample over time, where characteristics of the source sample would be inherited by the destination sample and any subsequent samples, so that the same data would not need to be entered repeatedly and only require change at the source sample to fix a mistake [130].

Coordination at the level of data exchange formats

Data formats are the containers used to transmit information from point of data entry or as output from an instrument into visualisation or analysis software or databases

(which may in fact be as simple as an archive of indexed files) [156]. The majority of modern data exchange formats are based on XML [156]. There has been a proliferation of standard data formats in the omic domain from the main standards developing groups [129]. The proliferation of independently created data standards makes the integration of data produced by different experimental techniques difficult, because the models differ in their representation of information common to all functional genomics experiments, e.g. in detail and terminology, so that semantically equivalent information is described in syntactically incoherent ways [129, 138].

In the next sections, the FuGE and ISA-TAB projects to define unified omic data formats are described.

FuGE Since its initiation by MGED in 2004 (now FGED to reflect focus on functional genomics standards development), the FuGE project had become a major cross-community collaboration, with developers from various organisations and representatives from industry and academia [129]. The aim of the FuGE project is to unify functional genomics data exchange formats for high-throughput experiments to create consistently structured data formats [129, 138].

The FuGE data model covers the description of biospecimens, protocols, instruments and software used [129]. FuGE was created essentially by removing the microarray specific parts of MAGE-OM and modifying it to enable the creation of technology specific extensions [129, 138]. The FuGE object model (FuGE-OM, [129]) models components that are universal to all types of functional genomics experiments across all technologies, e.g. preparation of sample, instruments, protocols, and multidimensional data [129, 138]. FuGE is comprised of 10 packages divided into two categories [129, 138] as shown in Figure 4.8. The “Common” package and its sub-packages are used for general data standards functionality [129, 138]. The “Bio” package and its sub-packages contain modules for describing experimental procedures, such as experimental design and sample annotation [129, 138]. FuGE can be used to encode biological workflows, such as experimental procedures or methods, SOPs, instrumental or software analysis procedures with the “Protocol” package [138]. The “Collection” package defines the root structure of the XML format and is used in an instance document to group all objects within a FuGE instance [223].

The FuGE-OM provides a data capture format (FuGE-ML) that can be used to in a stand-alone way to describe any workflow in biomedical science at arbitrary level of detail (depending on existence of appropriate ontologies), covering processes, data and materials, and comes with the facility to describe and reference other data sources, such as files, online data [147, 138].

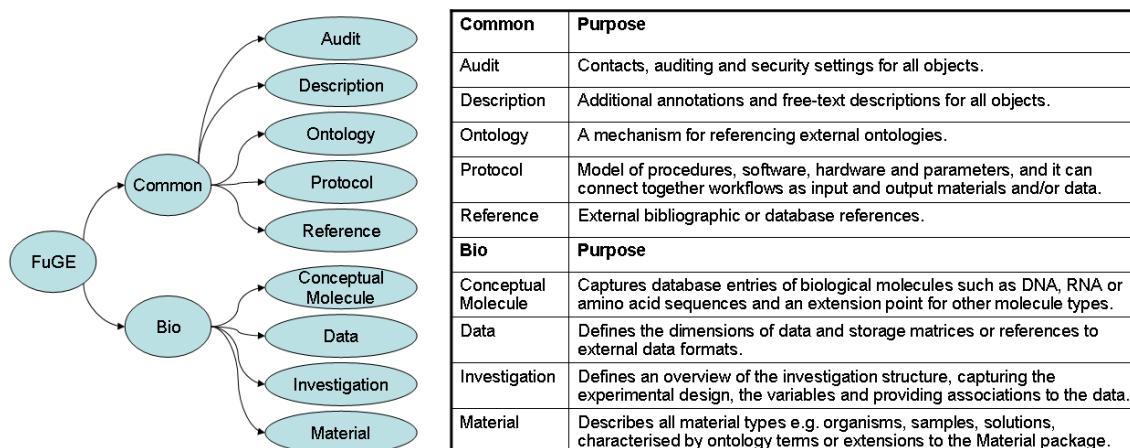


Figure 2.18: The hierarchy of base classes in FuGE. In addition to Common and Bio, a third package exists as a child of FuGE, Collection, which contains only classes and no child packages, Source: <http://fuge.sourceforge.net/>

FuGE has been designed so that it can be adapted for specific techniques (from [129, 138]):

1. through extension, either of the core FuGE model, with the specific model inheriting the functionality (such as auditing) and structure of FuGE or through development of ontologies for annotation
2. by a reference mechanism that can refer to external data formats (and as such embed them), such as other open or proprietary formats, while providing an accompanying metadata description

FuGE is intended to be used together with ontologies, such as FuGO [219] (later becoming Ontology of Biomedical Investigation (OBI)⁵², for the detailed annotation of data [129, 138]. FuGE can also be used to provide context to the technology-specific data recorded in established or proprietary data formats in an experiment, as these formats often lack the metadata that describes the context of their creation [138]. The Computational Proteomics Analysis System (CPAS) [224] project embeds established mass-spectrometry formats (such as mzXML [183]) and supporting information into FuGE to achieve the description of a complete workflow [138]. The PRIDE database supports PSI-formats that are FuGE extensions [134]. Both MGED and PSI have used FuGE as the basis for the development of data exchange formats [138], (from [129]), e.g. in

- MAGE-ML version 2 by the Microarray Gene Expression Data society (MGED),

⁵²OBI, <http://obi.sourceforge.net/>

a complete workflow format in an attempt to simplify the much more complex MAGE version 1 [129, 138].

- GelML (for gel electrophoresis), spML (for sample processing) and the gel informatics format, created by HUPO Proteomics Standards Initiative (PSI) [129, 147] and planned for analysis of mass spectra (analysisXML) [225, 138, 147, 147] and there are plans to “retrofit” mzData to the FuGE-OM to make it compatible [147]. mzIdentML is also based on the FuGE object model [202].

Many omic communities are considering harmonisation of their data formats with FuGE, such as PSI [153], or using FuGE for the development of domain specific data models, such as the Metabolomics Society and other standards development organisations are planning further FuGE-OM extensions [129, 138, 149]. FuGE has already been adopted by several omic communities in their development of data formats, such as the Genetical Genomics consortium in the MOLGENIS framework, XGAP [226], the Flow Informatics and Computational Cytometry Society [227] in the generation of FuGEFlow, an extension of the FuGE-OM specifically for flow cytometry data and metadata to support the Minimum Information about a Flow Cytometry Experiment (MIFlowCyt) standard [207]. The mark-up language, Flow-ML was generated from this data model, which can implement MIFlowCyt [227].

ISA-TAB The ISA-TAB format was proposed by the MGED RSBI as a common data format for capturing data and metadata from biological, biomedical, and environmental investigations, using various omics (such as genomics, transcriptomics, proteomics, and metabolomics) technologies together with more traditional methodologies [228]. ISA-TAB is a simple tab-delimited format, built around the Investigation, Study, Assay entities [155] for exchanging experimental metadata [228]. ISA-TAB is modelled on the successful MAGE-TAB format [168, 228]. Like MAGE-TAB, ISA-TAB files can be created using specific software or through spreadsheets, allowing easy creation, editing and viewing by researchers with little or no bioinformatics support [228]. ISA-TAB can be used as a simpler alternative for XML-based formats and as such competes with and duplicates efforts such as FuGE-ML [228].

Coordination at the level of Ontologies/CVs

The use of ontologies to annotate data has allowed complex data to be “semantically integrated” [155, 98]. CVs and ontologies facilitate in query functions in database searches, disambiguation and definition of synonyms [156]. Below, the FuGO/OBI project and the OBO initiative are presented, both of which are attempts at unifying and harmonising amongst biomedical ontologies.

FuGO/OBI The efforts for harmonising in ontology development started with the development of the Functional Genomics Investigation Ontology (FuGO). FuGO is an international collaborative project to provide terms for the unambiguous annotation of functional genomics studies, including the annotation of study design, materials, experimental protocols and instruments, created data and its statistical analysis [219, 219, 147]. FuGO was developed by extending the microarray specific MGED ontology (MO) [162], to become more generally applicable so that it could also be used to annotate data in the other omic domains, such as proteomics and metabolomics [219]. FuGO provides both terms common to all functional genomics studies, and domain specific terms and refers to existing established ontologies in annotations to avoid duplication [219]. As such, FuGO acts as “semantic glue” to facilitate that data from the various functional genomic sources will be understood the same way everywhere [219]. FuGO was later expanded to cover epidemiological and clinical research, biomedical imaging and other experimental research domains and eventually resulted in the Ontology for Biomedical Investigations (OBI)⁵³ [98]. The OBI is an ontology with a broad scope for annotating a large variety of experimental data in clinical and biological investigations [98]. The OBI ontology covers experimental designs, protocols, instruments and materials used in the investigation, processes, data and analysis types in all domains of biological and biomedical research [98]. Figure 2.19 shows the how a biomedical investigation may be represented using OBI Core, which contains a number of key terms from OBI and related ontologies.

Twenty-five groups are participating in the creation of OBI⁵⁴ under the umbrella of OBO Foundry [98].

The OBO Foundry The Open Biomedical Ontologies (OBO) Foundry is a community effort that promotes coordinated reform of existing and OBO ontologies, such as the GO and coordination in the development of new ontologies, to support data integration in the biomedical domain [98]. Over 60 groups have registered with the OBO Foundry [98, 229]. The OBO website⁵⁵ also acts as a public repository for other existing biomedical ontologies of different domains, such as anatomy, experiments, phenotype etc.

⁵³OBI, <http://obi.sourceforge.net/>, http://obi-ontology.org/page/Main_Page

⁵⁴OBI, <http://obi.sf.net/community>

⁵⁵OBO, www.obofoundry.org

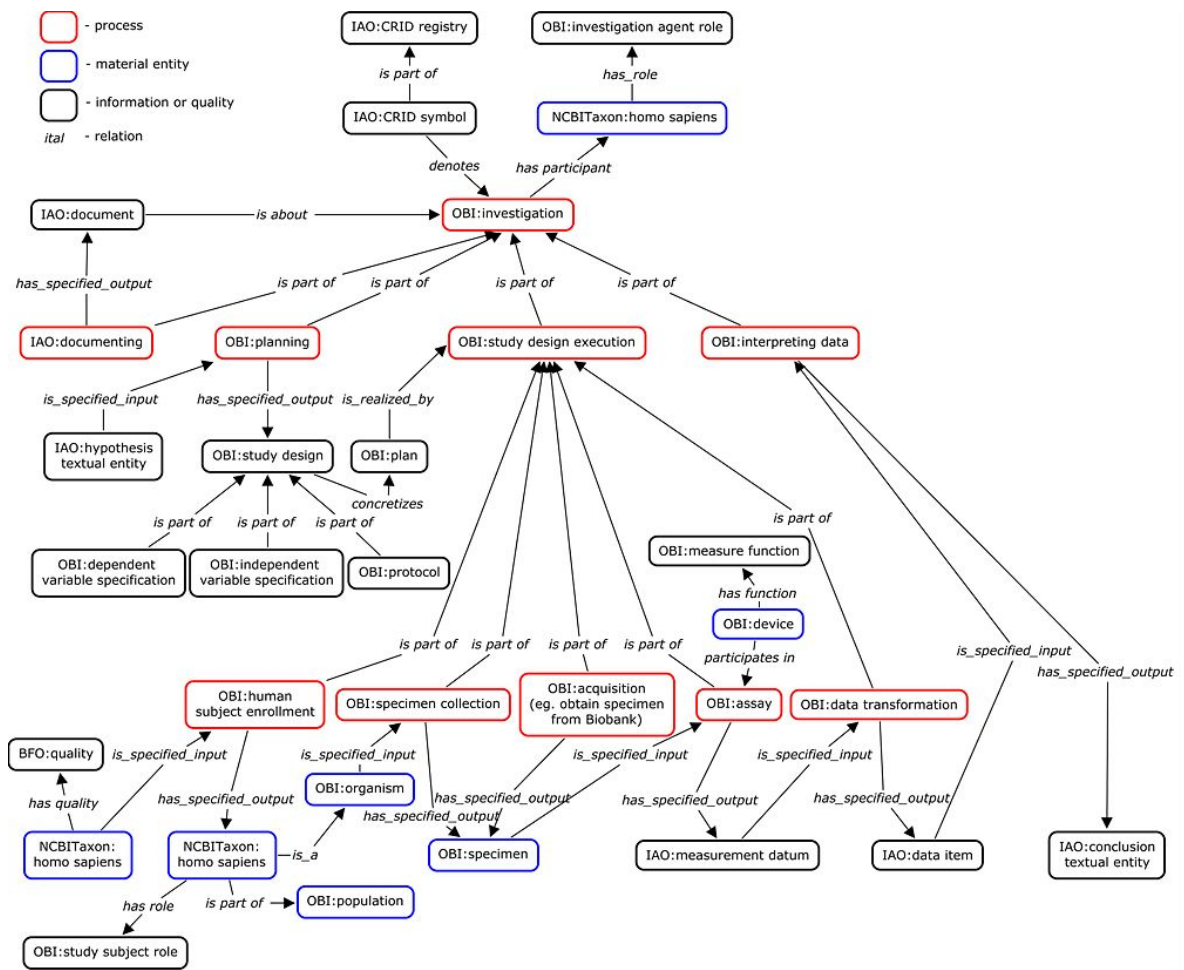


Figure 2.19: Representation of biomedical investigation using OBI Core, Source: <http://obi-ontology.org/page/Core>

2.4.8 Summary of omic data sharing

The development of omic reporting standards has gained momentum since the first major omic reporting standard, MIAME and many subsequently developed standards followed the same principles as or were modelled on MIAME [152, 155]. Modern omic reporting standards comprise three parts: MI specifications, data formats and controlled vocabularies [156]. MI specifications describe what information should be recorded, CVs or ontologies are used for annotating content in an unambiguous way to preserve meaning (i.e. the semantics) and standards-compliant data exchange formats are used for data transmission between tools and databases [156, 230]. A general characteristic of the development of omic data standards is a separation between the minimum information specification and the data format, with the data format capable of recording an unlimited amount of further information or even less information than specified in the minimum information specification [149]. Modern omic data exchange

formats are mainly based on XML [156]. To balance stability with dynamic development, omic reporting standards use ontologies or controlled vocabularies in conjunction with data formats: data standards can be kept stable to facilitate software implementation while the accompanying controlled vocabulary could be frequently updated to support new technologies, instruments, and methods of acquiring data [153]. The use of CVs or ontologies also allows for reasoning to be performed over the data.

Omic reporting standards have all been developed through community-based grass-roots initiatives, in a “bottom-up” way by communities of omic domain experts and stakeholders [128]. The argument is that while the idea of “perfect standards” is nice, ‘quick and dirty’ community-developed standards are better in a fast-evolving field like systems biology than slowly developed and rigid standards developed in a top-down approach that cannot easily adapt to changes in the field and thus become unsuitable as a field matures [231].

This stands in contrast to the standards developed in the domain of electronic health records, which are created top-down and involving major standards development organisations, such as ISO and CEN. The success of the omic community initiatives is evident in the wide adoption of omic standards in journal publications, databases, tools and software. This may be due to the fact that omic data reporting standards have traditionally been developed independently for each technology (such as for mass spectrometry or microarray analysis) and domain (metabolomic and genomic standards). In contrast, EHR standards attempts to encompass the clinical domain as a whole, which is a much more complex task due to the enormous field and thus will take longer to define and adopt and runs the risk of becoming outdated before it has been finished.

However, there are parallels between the development of omic data formats and archetypes. Omic data formats are file formats that model the information to be reported according to the omic data standards and can be used to safely exchange this information. Modern omic data formats are often based on XML, described with the XML schema language, such as mzML. The XML schema is used to express a set of rules to constrain the structure, content and semantics of XML instance documents that conform to the schema in a format that is both human-readable and machine-readable⁵⁶. Similarly, archetypes are used to express a set of rules to constrain the structure, content and semantics of instances files that conform to the archetype in a format that is both human-readable and machine-readable. However, archetypes have to adhere to additional semantic requirements with regard to the scope and structure of their content. Data formats do not have such restrictions and as such there is much greater freedom in how information can be organised in data formats than when modelled with archetypes. It is hypothesized that the data formats may be exploited

⁵⁶XSD, <http://www.w3.org/XML/Schema>

for the development of omic archetypes.

Omic data formats are consensus-based and the result of community initiatives involving domain experts. Similarly, to make archetypes widely applicable, they should be the result of a consensus-based approach involving a number of different domain experts [67, 68]. However, the development of clinical archetypes lags behind the development of omic data formats, possibly due to the higher complexity involved in the development of archetypes and the existence of other competing EHR standards.

2.5 Summary

This chapter provided the background to this thesis. First, standardisation efforts in the clinical domain to enable the safe sharing of EHR data have been presented. The safe sharing of any type of data means that it needs to be modelled in a consistent manner. The 2-level modelling approach has been proposed as a novel approach to model data in the clinical domain, which is characterised by its size, complexity and evolution. An insight into biobanking activity has been given, which has been rising since the development of high-throughput technologies, and demonstrating the need for data sharing within and between collaborating biobanks. Finally, efforts to enable for the safe sharing of data in the omic domain have been described.

The purpose of this thesis is to investigate the extent to which the *openEHR* approach can be used to develop the proposed eBMRR. Biobanks are at a crossroads between clinical patient and research information. They gather clinical patient information from the patient's medical record and create new research information through analysis of patients' biosamples stored in the biobank. The *openEHR* approach has been identified as a possible candidate for the development of this eBMRR, due to the 2-level modelling approach that may be applied to the omic domain, leveraging the multitude of omic data standards and formats, to develop archetypes that allow for the safe sharing of research information on patient samples in addition to the clinical information of the sample donor in a biobank context. The development of clinical archetypes is an ongoing process, so part of this investigation evaluates whether archetypes developed for the clinical context can be re-used to describe clinical patient information in a biobank context. The approach taken to carry out this investigation was to take the database schema of an existing biobank and attempt to model the information contained in it with (existing) *openEHR* archetypes. The process and outcomes of this attempt are described in the next chapter.

Chapter 3

Modelling the database of a biobank information management system with archetypes

3.1 Introduction

To support biomedical knowledge discovery, clinical patient data needs to be integrated with research information on patient samples. Biobanks combine these two streams of information and a biobank information management system stores both clinical patient information and information related to the patient samples, often in a proprietary way, thus making it difficult to share the information and making it available for future research. However, the information created by carrying out research on biosamples is not generally integrated with the clinical patient data and thus not easily available for future integrative and longitudinal research. Researchers store omic information created through analysis on patient samples on the researchers' computers, often in a number of incompatible data formats, or they may be uploaded into one of the many emerging omic databases, e.g. to fulfil requirements for a research publication. In summary, research data created on the patient samples is not integrated with the clinical information in the patient's (longitudinal) health record, but may in the best of cases be recorded in a number of omic databases.

The eBMRR has been proposed as a new type of record that can generically integrate both clinical patient data and research data, such as omic data won through high-throughput analysis on patient samples, facilitating the safe sharing of information and making it available for future research, ultimately to support biomedical knowledge discovery.

The *openEHR* approach, developed to support the creation of future-proof, semantically interoperable EHRs has been identified as a possible approach to create this new type of record. Since biobanks combine the two streams of information, clinical patient data and research information created through scientific experiments on patient

samples, the eBMRR will be developed in a biobank context in this research. However, basing the development of the eBMRR on an EHR standard will also enable the wider generic integration of information with other EHRs that adhere to the same standard, outside the context of a biobank. Using the *openEHR* approach may also reduce development time for biobank information management systems due to the genericity of archetypes, allowing them to be reusable.

The development of clinical archetypes is an ongoing process. Thus, as a first step towards the creation of the eBMRR, this chapter is concerned with the question whether *openEHR* archetypes developed for the clinical context can be re-used to model the clinical patient information in a biobank context.

To answer this question, the following experiment was carried out: the database of the BIMS in an existing biobank, namely the biobank of the Irish Prostate Cancer Research Consortium (PCRC)¹, was taken as the basis for modelling the information contained in it with (existing) *openEHR* archetypes. Another aim of this experiment was to explore the feasibility of making the data in the database of the PCRC BIMS available through archetypes, either through an archetype layer above the existing database or by migrating the existing data into an archetype-based database. The process and outcomes of this experiment are described in the following paragraphs and an evaluation of this experiment is provided in chapter 5. This research has been presented at the *14th Annual Conference of the Health Informatics Society of Ireland (HISI)*, where the paper received the “Best Paper Award” and published in the *International Journal of Medical Informatics* [232]. This thesis presents an extended and updated version of the research described in the paper.

3.2 The PCRC biobank

To investigate how the eBMRR could be realised in a biobank setting, a collaboration with a biobank, namely the Irish PCRC, was established. The aim of this collaboration was to investigate how the eBMRR could be set up in a biobank to support the recording of both clinical and research information by modelling the information stored in the BIMS with archetypes and templates, to enable the safe sharing of this information and making it available for future research.

The PCRC was set up in October 2003 and is a multi-disciplinary, trans-institutional collaboration between several Dublin-based research institutions and hospitals, including University College Dublin, Trinity College Dublin, Royal College of Surgeons and Dublin City University. It aims to bring researchers together to improve diagnosis and treatment for prostate cancer. The Consortium partners with several Dublin-based

¹Prostate Cancer Research Consortium, <https://pcrc.tchpc.tcd.ie/>

research institutions and hospitals, including The UCD Conway Institute, St. James's Hospital, Mater Misericordiae Hospital, The Trinity College Dublin Institute of Molecular Medicine, The Royal College of Surgeons in Ireland, and Beaumont Hospital.

The main research aims of the PCRC is the discovery and validation of novel prostate cancer biomarkers for disease diagnosis, prognosis and treatment success and to understand the molecular basis of the disease and its recurrence, as well as the development of new therapies. Central to the PCRC is the longitudinal collection of biospecimens of prostate cancer patients. The PCRC collects samples of prostate tissue, urine and blood from several hospitals. The prostate tissue arrives after clinical pathological analysis has been carried out on it. These samples are further processed, e.g. tissue is preserved and/or embedded, urine is treated with a protease inhibitor and blood is separated into plasma and serum, following standard operating procedures (SOPs), before being stored in the biobank's freezers.

The bio-resource currently holds over 350 tissue samples with matched serum and plasma samples as well as over 90 urine samples and contains comprehensive clinical information about the patients and their disease progression. For their biobank information management system, the PCRC uses the proprietary product "Distiller", developed by the Slidepath software company², an online laboratory and clinical-data management platform [233]. Distiller can be used to store project information, tissue microarrays, digital slides and other multimedia in hierarchical, fully searchable databases [234]. The software provides an easy-to-use (PHP-based) point-and-click interface that enables non-technical users to create their own database table definitions without having to know any database query languages or the underlying relational database implementation [234]³.

The flexibility of this system means the information in a Distiller database cannot easily be shared, since the database structure will be different between systems that may be set up by different users without collaboration. Each biobank that uses Distiller may define the data structures for the information to be stored according to their needs in an ad hoc fashion, possibly in a completely different way to other users storing essentially the same information. Furthermore, the database does not contain meta-data on top of table and field names, which means that this database by itself is semantically inaccessible for further use and prohibits the integration of the information with medical records. Thus, if the information stored in the database tables developed by the PCRC researchers could be modelled with *openEHR* archetypes and templates, then these may be used in the future to augment the original database with an archetype layer or the legacy data migrated into a new archetype-based database to

²Slidepath, www.slidepath.com/, now part of Leica, www.leica-microsystems.com/

³<http://www.leica-microsystems.com/products/digital-pathology/manage/details/product/distiller/>

provide the necessary context to the data to allow for it to be shared safely and stored in a future-proof way.

As a first step, the information recorded in the proprietary database of the prostate cancer biobank of the Irish Prostate Cancer Research Consortium (PCRC) needed to be modelled with *openEHR* archetypes and templates. In other words, the task was to find ways to reconcile the information structures of the PCRC BIMS as developed by the PCRC researchers with *openEHR* archetypes and templates so that the information in the BIMS could be accessed and shared in a safe, contextualised manner. This experiment will provide an insight to what extent existing clinical *openEHR* archetypes can be re-used in the context of biobanks to model the clinical concepts embedded in the database of a biobank.

3.3 The meta-data of the PCRC biobank information management system

The PCRC provided a conceptual database schema that reflects how the information in the PCRC BIMS is stored in the various tables created by members of the PCRC in the PHP interface. The tables that are created by the users through the PHP interface do not correspond directly to the tables that the software creates in turn in the underlying relational database. A proprietary mechanism is used to link the front-end to the back-end in a user-friendly manner, in a bid to allow the creation of tables by users without database knowledge.

The conceptual database schema, rather than the underlying database schema, provides the semantics of the data by showing the table-names, field-names and hierarchical relationships between tables and is used as basis for modelling the concepts embedded in the database of this biobank.

This conceptual database, shown in Figures 3.1 - 3.4, contains 18 interlinked, conceptually hierarchical tables, in a patient-centred hierarchy (i.e. the description of the patient is at the root of the table hierarchy), with 146 data fields in total. The tables are interlinked through the use of an identity field in each table, that has no external semantic meaning (except for the ID in root “Patient” table), thus reducing the number of information-bearing fields to 129. The database is used to record the following information:

- clinically relevant patient background information, such as past and present medical history,
- information detailing current medication, family history and current symptoms,

- information about patient treatment options and outcomes,
- information about specimen collection and storage, i.e. for prostate tissue, urine and blood,
- specimen processing information,
- results of simple histo-pathological tissue analysis,
- and the results of biochemical tests that have been carried out on the specimen as part of the diagnosis and which are documented in the patient's health record.

The PCRC BIMS is not currently set up to store the results of secondary analysis of specimen for research purposes, such as the execution and subsequent results of omic experiments, which each researcher currently stores individually on his or her computer and in arbitrary formats (for example in the proprietary format of a machine output). The biobank has voiced the intention to be able to make this research information created by the researchers from experiments on patient samples shareable and available for future research.

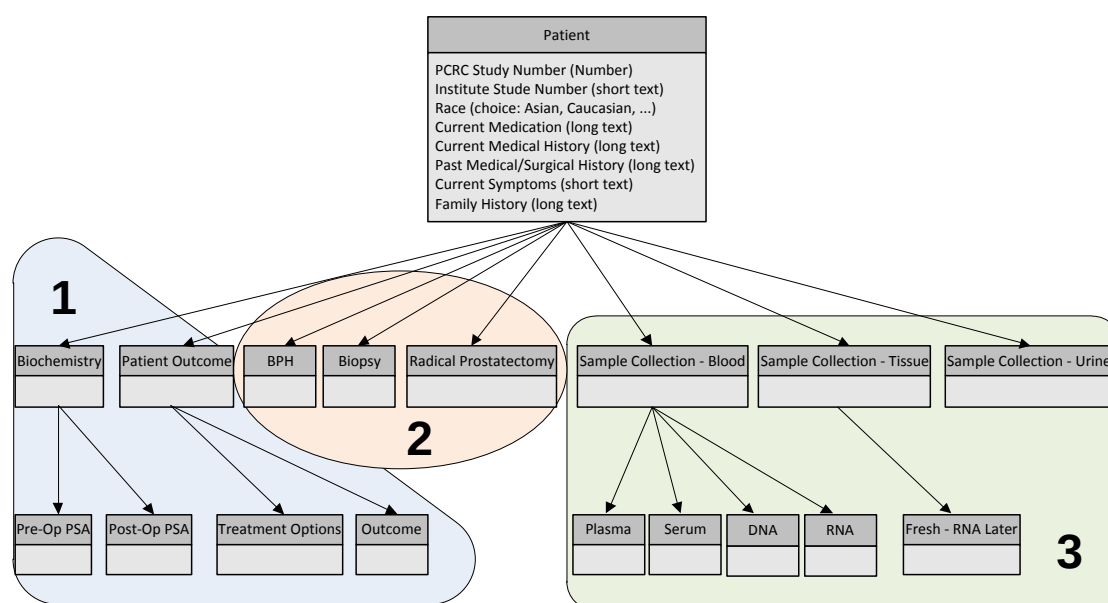


Figure 3.1: Overview of the hierarchical PCRC BIMS database table relationships - the three parts are shown in detail in the next three figures

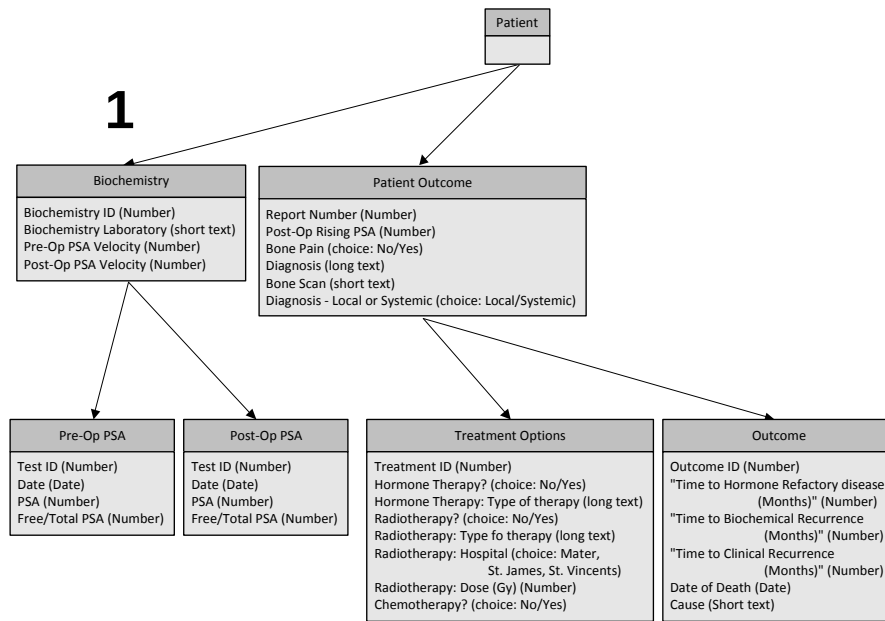


Figure 3.2: Part 1 of the PCRC BIMS database

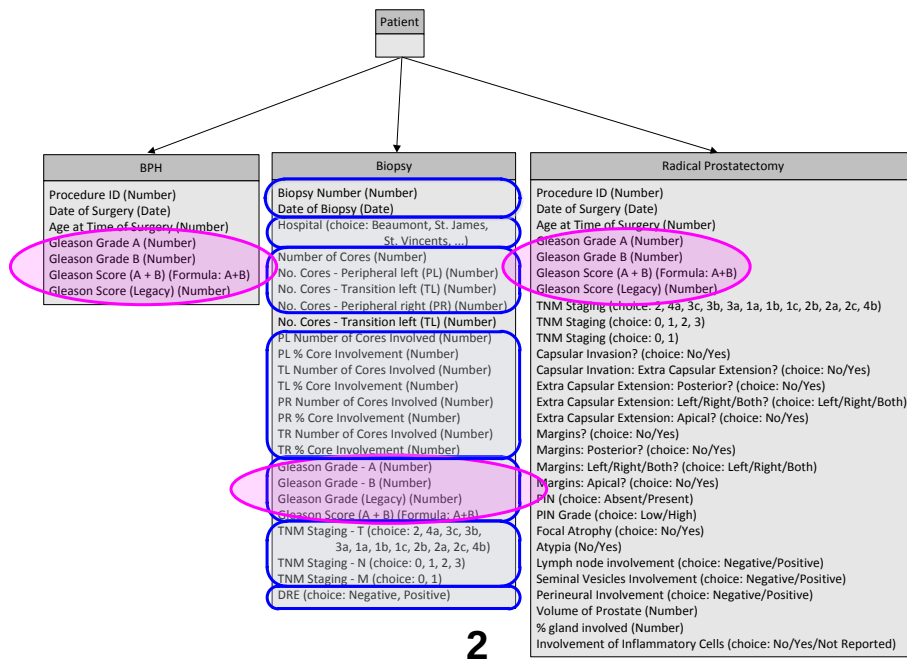


Figure 3.3: Part 2 of the PCRC BIMS database; the meaning of the pink ovals and blue rounded rectangles is explained in Section 3.5.

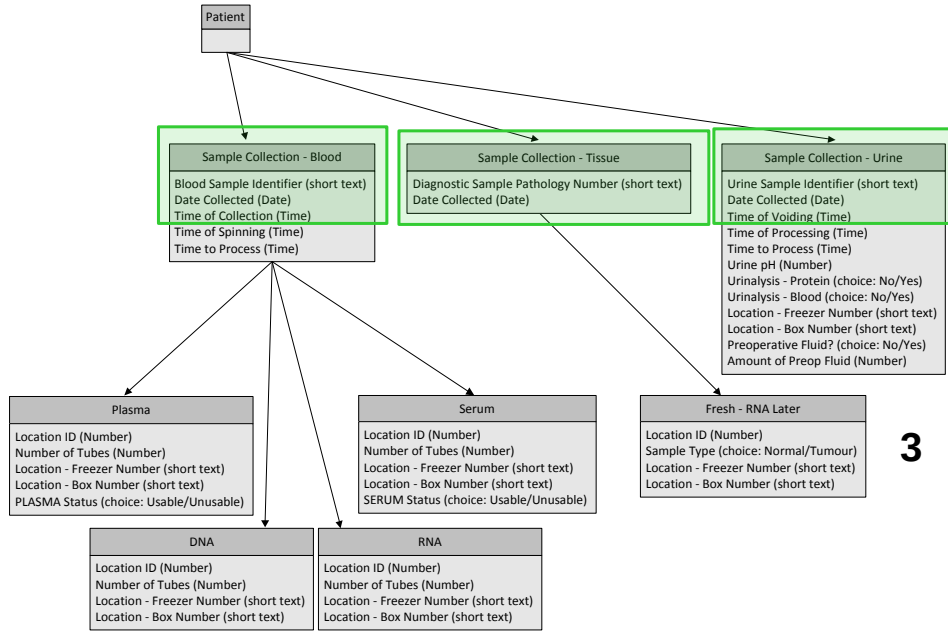


Figure 3.4: Part 3 of the PCRC BIMS database; the meaning of the green rectangles is explained in Section 3.5.

The provenance of instance information entered into the database can be classified as follows:

- Exact copies of defined pieces of the original clinical data found in the patient’s chart in the hospital (those pertinent to the research carried out by PCRC associated researchers), emulating the information in the patient chart,
- summary information created through derivation of original clinical data found in the patient’s chart that is kept in the hospital,
- information gathered from patient interviews or copied from proformas filled out during the consent process,
- and information created as part of the sample collection and processing procedures.

Information is entered into the database from the patient charts/proformas etc. by the research nurses. For follow-up data the hospital or GP is contacted. Tables may be instantiated several times in the same hierarchy. This means, for example, that when a new sample is collected, a new table to describe the particular sample collection event is instantiated. Instances of tables may be updated, so for example, if the number of tubes available for a sample changes due to the sample being used in a study, then the table data is updated. Some tables will only ever be instantiated once in a particular patient hierarchy, for example the “Radical Prostatectomy” table, since

this is an event that will only happen once. Similarly, the “Treatment Options” table is only instantiated once and then data may be updated in it, but generally, no more than one “Treatment Options” table is instantiated per patient. The table hierarchy is reflected in the interface of the Distiller software, which allows users to click through the instantiated table hierarchy of a patient.

The information in the PCRC biobank enables the data-driven analysis on the content of the PCRC BIMS database over a large set of prostate cancer data. It is further used for the selection of samples with specific characteristics or where the patient fulfils specific characteristics for (further) analysis, such as in omic experiments. For example, it may be used to select prostate cancer patients with a certain characteristic, such as a specific age range or a certain Gleason Score recorded from pathological analysis of the resected prostate cancer tissues.

14 of the tables contain mainly information related to a patient sample (such as information regarding its collection and results of pathological analysis), whereas the remaining 4 tables contain mainly clinical information. With regard to instance data, the database contained data for over 500 patients. The data types of the content were of type text, boolean, numbers, e.g. float or integer, or date/time. Some of the text fields contained either unstructured text (for example for medical history and current symptoms), or the result of choosing a text item from a list (such as Caucasian, Asian etc. for “Ethnicity”). In many instances, table data was incomplete with empty fields and there was no instance where data was entered in all available fields for one patient.

Archetypes and templates provide context around data. The PCRC database schema by itself does not provide much contextual information and it is not always obvious what the exact meaning, context and purpose of each table name or field is. For example, table names are often misleading and not always indicative of the information stored in it. For example, most of the information stored in the table called “Radical Prostatectomy” covers the results of pathological analysis of the prostate specimen, with only one field covering details of the radical prostatectomy operation itself (“Date of Surgery”). Thus, to be able to map the content of the PCRC database correctly to existing archetypes, the exact meaning and context of the table and field descriptions needed to be established. To find out this information, the research nurses who populate the database and who have been involved in the design of the database schema were consulted. However, the research nurses had only just started to document this information. Thus, much of the meaning and context of the data collected in the PCRC database was gathered through analysis of the table names, field names and field contents in consultation with available literature on practices and artefacts of the medical domain in general and prostate cancer in particular. Further interviews with the research nurses were carried out to clarify the meaning of the outstanding data

PCRC BIMS data type	<i>openEHR</i> data type
Date	DV_DATE
Number	DV_ORDINAL/DV_PROPORTION/ DV_QUANTITY/DV_COUNT/DV_INTERVAL
Number/text, semi-structured	DV_ORDINAL/DV_PROPORTION/ DV_QUANTITY/DV_COUNT/DV_INTERVAL or DV_TEXT/DV_CODED_TEXT
Text, choice	DV_CODED_TEXT
Text, unstructured	DV_TEXT
Time	DV_TIME
Time interval	DV_DURATION

Table 3.1: Mapping of the BIMS data types to *openEHR* data types

items and the data that is to be entered into it, and these discussions fed back into the nurses’ documentation for the use of this biobank. A table detailing the meaning of the database tables and fields is provided in Appendix A. In addition, further information is created on a meta-level that is not reflected in the PCRC database schema. In particular, this concerns sample collection: when entering data about a particular sample being collected, there is an option to select at which event the collection took place, either at the transurethral resection of the prostate (TURP, marked as “BPH” in the PCRC BIMS database), biopsy or radical prostatectomy. In an archetype-based record system, it is the knowledge assumed by the users of the PCRC BIMS that is being recorded by the nurses in external documentation that would be encoded in archetypes and templates. It was necessary to return to the process of information refinement throughout the modelling process to make sure that the mappings between the items in the archetypes and the fields in the BIMS database schema were correct.

3.4 Mapping of data types

First, the data types in the BIMS database needed to be mapped to the data types of the archetype information reference model to see if the data types specified in the *openEHR* reference model could be sufficient to accurately represent the BIMS data types.

The reference model assumes a number of primitive data types (namely Integer, Real, Boolean, Character, Octet, String, List<T>, Set<T>, and Array<T>, described in the *openEHR* Support Information Model), which build the basis for higher level data types in the *openEHR* Information Reference Model, which are all subclasses of the *openEHR* class DATA_VALUE and pre-fixed with “DV_”. It was found that the *openEHR* reference model could sufficiently cover all data types that appeared in the PCRC database schema. The mappings are shown in Table 3.1.

There also exists an *openEHR* data type, DV_DATE_TIME, which can combine both the date and time description of an event, e.g. the two fields “Date Collected (Date)” and “Time of Voiding (Time)” in the “Sample Collection - Urine” table could be combined into one archetype field of data type DV_DATE_TIME, that could be called “Date and time of urine collection”. After it was determined that the *openEHR* data types would be sufficient to model the data types in the PCRC BIMS, the next step was to identify archetypes that would cover the concepts in the BIMS database.

3.5 Semantic mapping

To model the data in the PCRC biobank with *openEHR* archetypes, the following process depicted in Figure 3.5 was applied.

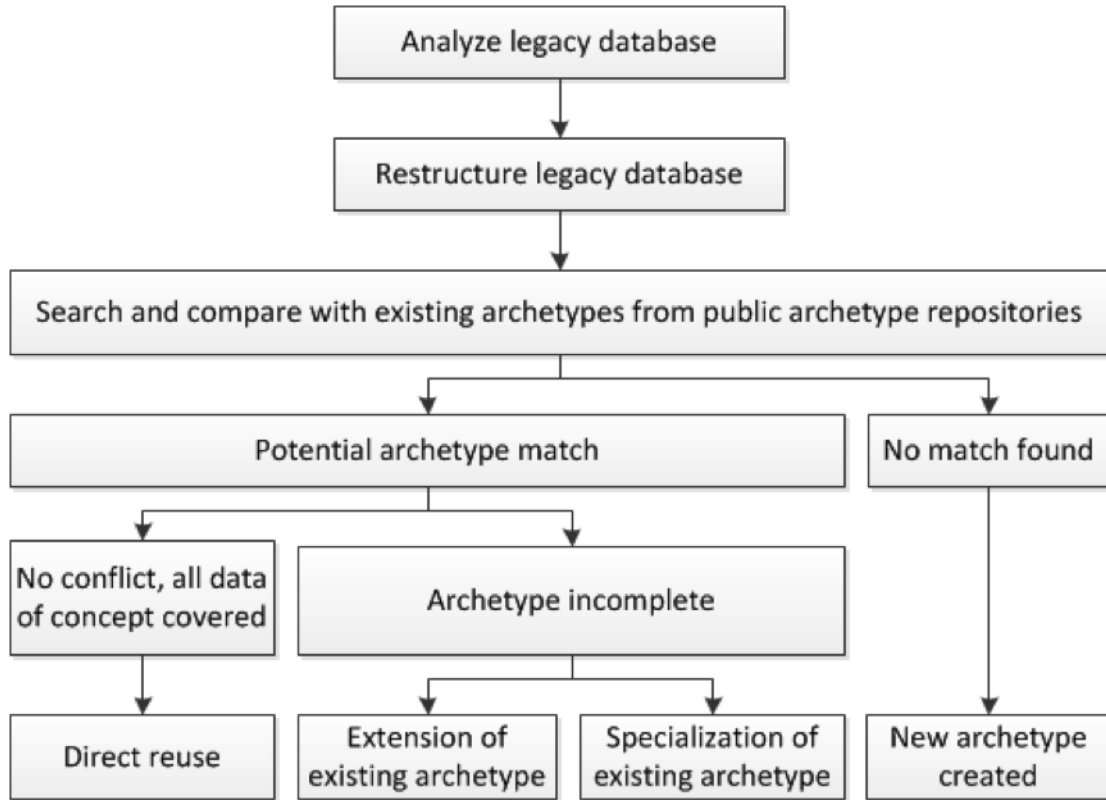


Figure 3.5: Flow-chart of modelling the data in the BIMS database with archetypes

The set of archetypes that result from the application of this process then need to be arranged into templates to reflect the specific data recording requirements of the PCRC biobank.

While this research was carried out, Buck et al. [57] published a paper to model the data in a neonatology electronic patient record (EPR), which essentially proposes the same approach. In fact, the flow-chart shown in Figure 3.5 corresponds to steps two to four of their five-step *openEHR* data modelling approach *odma*.

The five steps of *odma* (from [57]):

Step 1: Determine all items

Step 2: Merge items into concepts

Step 3: Map the derived concepts to existing archetypes

Step 4: Develop archetypes

Step 5: Design templates

The PCRC database provides all the items that are currently being stored in the database of the PCRC BIMS, so the *odma* Step 1 is already fulfilled. A detailed analysis was carried out to determine the exact meaning and information context of each item as described above. The design of templates from the resulting set of archetypes defined in this process, corresponding to *odma* Step 5 is described separately in Section 3.7. How the proposed process was applied in the context of the database of the PCRC biobank are now described in more detail.

To consider how the data in the PCRC BIMS may be modelled with archetypes one has to bear in mind that each particular archetype models one distinct concept [37]. An examination of existing archetype repositories revealed that the organisation of the data in the conceptual BIMS database could not be used directly as a basis for identifying suitable archetypes, since it groups data in a completely different manner compared to that used in archetype design (based around concepts).

An analysis of the conceptual BIMS database schema showed that often the tables in the BIMS database contain a number of distinct concepts. For example, the Biopsy Table contains information about the conditions surrounding sample procurement, a description of the sample, the results of pathological analysis, an evaluation of those results, clinical cancer staging information, and the findings of a physical examination of the patient, as highlighted in Figure 3.3 with blue rectangles. On the other hand, some data structures were found to be used repeatedly in different tables. For example, the fields related to the Gleason Staging system⁴ appear in three different tables, as highlighted in Figure 3.3 with pink ovals. Similarly, there are three tables concerned with sample collection: Sample Collection Blood, Sample Collection Tissue and Sample Collection Urine, as marked in Figure 3.4 with green rounded rectangles, two of which model virtually the same information related to sample processing such as “Date Collected”, “Time of Voiding” or “Time of Collection”, “Time of Spinning” or “Time of Processing” and “Time to process”, with the third table modelling a subset of the former two.

Other tables also contained very similar types of information, for example, the tables describing blood sample derivatives (Plasma, Serum, DNA and RNA) each contain

⁴The Gleason Staging system is a used to grade the severity of prostate cancer. The pathologist assigns Gleason grades to the tumour patterns found in microscopic analysis of the prostate cancer tissue. The Gleason score is the sum of the individual grades.

sample location information and the number of tubes that contain the sample. Additionally, information related to the same concept may be spread over several tables. For example, the results of the clinical histo-pathological analysis of prostate tissue are recorded in the BPH, Biopsy and Radical Prostatectomy tables. Thus, to facilitate the next step, i.e. searching for and identifying suitable candidate archetypes from public repositories and the creation of new archetypes, the tables and fields in the BIMS database schema were first regrouped into concepts. This task basically corresponds to *odma* Step 2: “Merge items into concepts”, except that the merging is achieved by reorganisation of the data. This step helps avoiding that one concept is modelled repeatedly in different archetypes [57].

Thus, as part of this regrouping step, the number of items to be modelled could be reduced, such as the information regarding sample storage information, such as “Location - Freezer Number” and “Location - Box Number” which appeared in five tables and the four fields involving the histo-pathological cancer staging tool Gleason Score, which appeared in three different tables. An archetype modelling the concept Gleason Score could then be reused in different contexts, e.g. to describe the Gleason Score of Biopsy Cores, or to describe the Gleason Score of resected tissue, as is required in this biobank, through the slot mechanism.

This reorganisation resulted in 23 high-level concepts. During this reorganisation process, it was noticed that some of these concepts were either centred around patient (such as past medical and surgical history, family history, Cancer treatment, etc.), whereas others were centred around the sample, such as details regarding sample collection, sample processing and Laboratory test results. However, sometimes this distinction could not be made, where a concept combined both information related to the patient sample and the patient itself. For example “TNM Cancer Staging”, here included under the sample-side, is a cancer grading system that combines the results of analysis of prostate tissue (‘T’) (either from the resected prostate or from a fine needle biopsy) together with results of analysis of metastasis to regional lymph nodes (‘N’) or analysis distant metastasis (‘M’). Taking this observation into account, the 23 concepts were divided into 13 clinical patient-related (left side of mind-map) and 10 sample-related (right side of mind-map) concepts (see mind-map showing high-level concepts in Figure 3.6 and in detail in Figure 3.7) with several specifications. The annotations in Figure 3.7 correspond to the annotations in Figures 3.1 - 3.4, and highlight how these fields have been reorganised. This reorganised schema was then used as the basis for the archetype modelling process.

The next step was to model the concepts in the database with archetypes, either reusing archetypes where suitable ones exist or creating new archetypes where none is available. A single archetype should represent a single clinical concept and archetypes

need to be designed in a generic and reusable way so that they can be used in different contexts to enable interoperability and thus the sharing of EHRs and EHR Extracts [37]. This also means that the proliferation of archetypes modelling the same concept should be avoided. Therefore, as recommended also by the *odma* approach, it is important to try and reuse as many existing archetypes as possible rather than creating an entirely new set of non-generic archetypes based on the particulars of this prostate cancer biobank, which would have only shifted interoperability issues to the level of archetypes rather than solving them. Trying to reuse existing archetypes where possible also avoids carrying out the unnecessary work of re-developing high-fidelity archetypes de novo as well as risking the development of overlapping archetypes that could jeopardise interoperability [69, 235, 57].

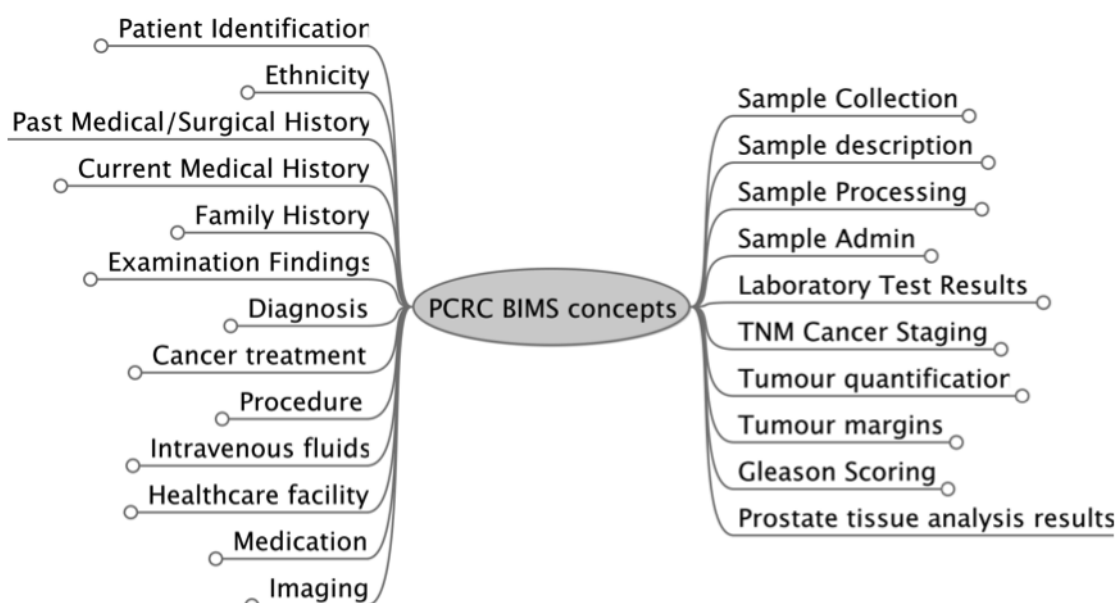


Figure 3.6: Re-grouping of BIMS concepts (leaf nodes collapsed) (left: clinical patient-related concepts, right: sample-related concepts)

It was hoped that archetypes covering the concepts in the PCRC BIMS had already been developed, since almost all of the information currently recorded in the PCRC BIMS has been gathered from the patients’ health record and clinical archetype development is ongoing in the *openEHR* community. At the same time it is important to note that the archetype approach is not a mature approach, so only a few hundred archetypes exist, most of which are still in draft form. For example, the NHS archetype repository contains only 1726 archetypes (15/01/2010), of which 966 are of lifecycle_state = <“0”>, 502 are of lifecycle_state = <“Initial”> and the remaining 222 archetypes are of lifecycle_state = <“AuthorDraft”>, 32 archetypes are of lifecycle_state = <“OrganisationDraft”> and 4 archetypes are of lifecycle_state = <“NotSet”>. At the time of writing, most archetypes in the *openEHR* archetype repository are still in a

draft state (222 out of 244 active archetypes in total), with 15 in Team review state and 7 published (as of January 15th, 2010).

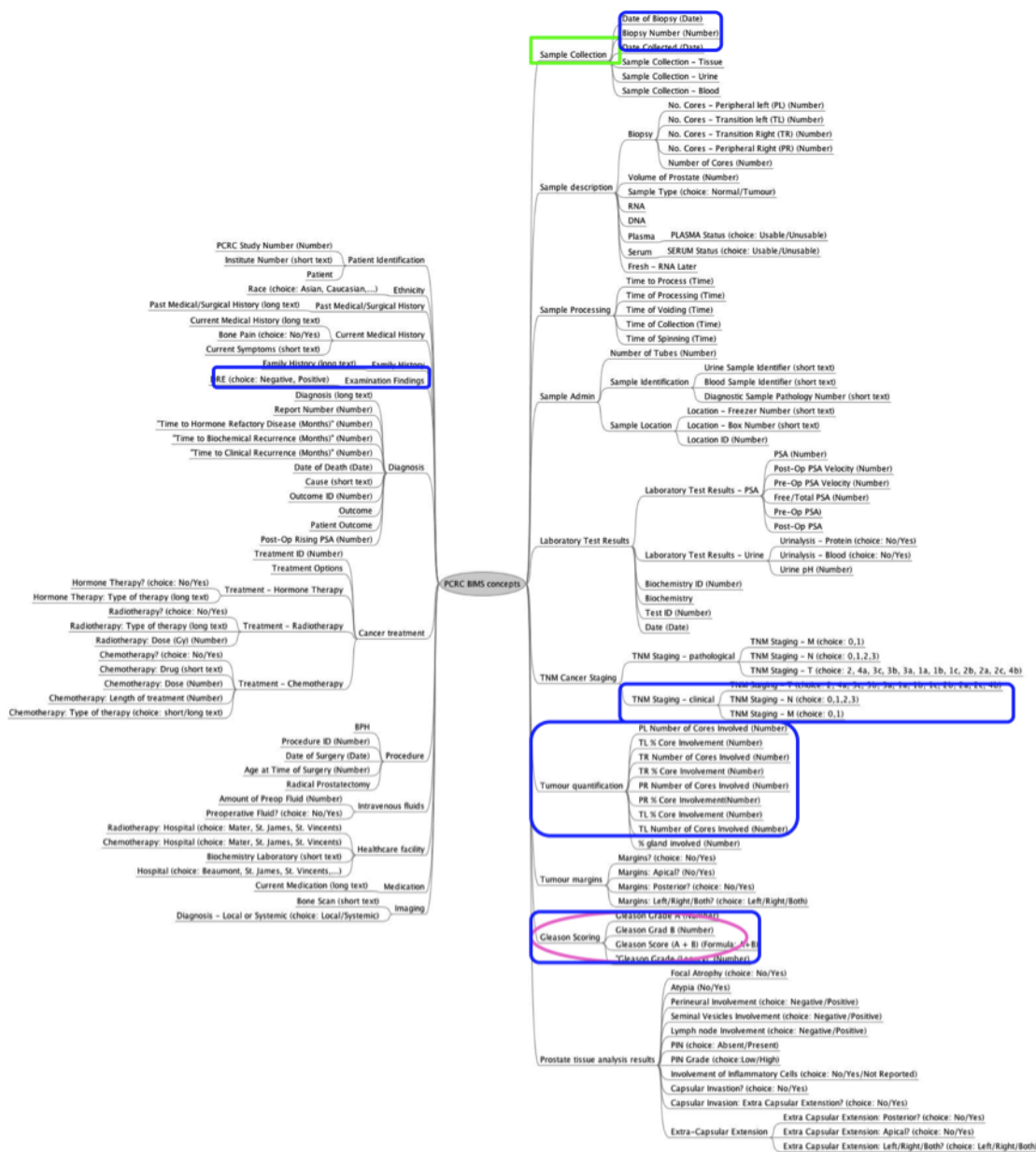


Figure 3.7: Re-grouped BIMS fields; the pink ovals, blue rounded rectangles and green rectangles correspond to those in Figures 3.3 and 3.4.

Two archetype repositories, the *openEHR* archetype repository⁵ and the NHS archetype repository⁶ were searched for candidate archetypes. The *openEHR* online repository was searched using the available string search function or through manual inspection of the archetypes, using the mind-map representation provided or looking at

⁵*openEHR* archetype repository, www.openEHR.org/knowledge/

⁶NHS archetype repository, <https://svn.connectingforhealth.nhs.uk/svn/public/nhscontentmodels/TRUNK/cm/archetypes/adl/openEHR/ehr/>

the ADL files directly. The latest NHS repository was downloaded and searched both manually by visual inspection of the ADL files and also through string search using the command line. Although the *openEHR* repository is a much smaller repository than the NHS one, the authors consider the archetypes of the CKM to be of higher quality because *openEHR* promote a consensus-based archetype development process involving teams of domain experts [236]. Thus, archetypes were primarily used from this source, but the NHS repository was consulted if no matching archetype could be found in the *openEHR* repository. However, due to partial overlap detected between archetypes in both repositories it was decided to mainly use archetypes from the *openEHR* repository, and only use archetypes from the NHS repository when no overlap was detected. There are many different ways to group information semantically, depending on someone’s viewpoint, so the regrouped information may not necessarily match how the information is grouped in archetypes and often several searches had to be carried out to find the information grouped into one concept.

When potential matches were found, these archetypes were analysed in detail to see if they modelled the information presented in the regrouped PCRC BIMS fields or if they were incomplete or if there was a conflict in how the information was modelled in the PCRC BIMS database and the archetype. In principle, each archetype should model a complete clinical knowledge concept and cover the maximum data set possible, and thus there should be no incomplete archetypes. However, the archetype approach is still quite new and good quality archetypes are the result of an ongoing consensus-based approach, so the number of stable archetypes is small. When an archetype was found to cover all fields of a certain BIMS concept, it was reused as is. When an archetype was found to cover only some of the fields in the regrouped BIMS concept, the repositories were searched again for further archetypes that covered those.

However, if no other archetypes could be found to model the information in these fields, the archetype was deemed to be incomplete and was extended to contain the missing items, as they belonged to the same concept. If an archetype was found that covered a more general concept than one needed in the BIMS, the archetype was specialised to the more specific BIMS concept. Finally, when no archetype could be found for certain fields of the BIMS database, then a new archetype was designed. These tasks match the tasks that [57] describe under *odma* Step 3: “Map the derived concepts to existing archetypes” and Step 4: “Develop archetypes”. Initially the Linköping Archetype Editor⁷ and later the Ocean Informatics archetype editor⁸ were used to review, modify, specialise and create new archetypes.

⁷LiU Archetype Editor, <http://www.imt.liu.se/mi/ehr/tools/>

⁸Ocean Informatics Archetype Editor, <http://www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-archetype-editor.html>

3.6 Archetypes modelling the data in the PCRC biobank

The application of the above process resulted in a set of 55 archetypes to fully cover all the fields in the biobank database (see Table 3.3). Many archetypes from the public repositories could be re-used without change in the context of this biobank, making up about half of the archetypes needed to cover the concepts in this biobank database (28 out of 55; 51%). However, not all fields of the PCRC BIMS database were covered by existing archetypes, so that five new ENTRY (ACTION, ACTIVITY, INSTRUCTION, ADMIN_ENTRY, OBSERVATION, EVALUATION) archetypes needed to be developed:

- openEHR-EHR-ACTION.radiotherapy.v1.adl
- openEHR-EHR-ADMIN_ENTRY.donor_admission.v1.adl
- openEHR-EHR-ADMIN_ENTRY.specimen.v1.adl
- openEHR-EHR-EVALUATION.age.v1.adl
- openEHR-EHR-EVALUATION.psa_velocity.v1.adl
- openEHR-EHR-INSTRUCTION.intervention.v1.adl
- openEHR-EHR-OBSERVATION.lab_test-psa.v1.adl

The PCRC BIMS provides data in a certain context and as such, the final organisation of archetypes and their hierarchical relationships need to be taken into account if an archetype-based record was to be developed for the PCRC BIMS. In an *openEHR*-based record, information is organised in Compositions, which act as the primary data container and is the unit of committal and communication of information [42]. That means that instances of ENTRY or CLUSTER type archetypes cannot be committed to the record system on its own, but that they always need to be recorded under the content attribute of a COMPOSITION archetype, so that a complete hierarchy from the COMPOSITION to the leaf data types is created. In practical terms, this means that one needs to think how the lower-level archetypes need to be arranged within the higher-level, organisational level archetypes, such as COMPOSITIONs and SECTIONs through the slot mechanism. Only when all necessary archetypes are present that can be used to model the data from the level of the composition down to the level of the data item can templates be developed that further constrain the information modelled by the archetypes to adapt them to the specific context of the PCRC biobank. In the process of finding or creating suitable organisational level archetypes, the high-level

table structure of the PCRC is taken as the model and adhered to as closely as possible, so that one COMPOSITION is used for each table in the PCRC BIMS database. In other words, the specific data model of the PCRC biobank can only be represented through the use of templates that combine and adapt the set of archetypes that cover all the concepts extracted from the PCRC BIMS to reflect the specific requirements of this biobank, rather than just through the generic archetypes alone. In the light of the above, the following new organisational archetypes were defined to provide a way to organise the archetypes that cover the data in the PCRC BIMS in a sensible way so that the meaning of the information recorded in the PCRC biobank can be preserved.

- openEHR-EHR-COMPOSITION.outcome_summary.v1.adl (persistent type)
- openEHR-EHR-COMPOSITION.patient_background.v1.adl (persistent type)
- openEHR-EHR-COMPOSITION.specimen.v1.adl (persistent type)
- openEHR-EHR-COMPOSITION.treatment_summary.v1.adl (persistent type)
- openEHR-EHR-SECTION.administrative.v1.adl
- openEHR-EHR-SECTION.chemotherapy.v1.adl
- openEHR-EHR-SECTION.current_medical_history.v1.adl
- openEHR-EHR-SECTION.current_medication.v1.adl
- openEHR-EHR-SECTION.current_symptoms.v1.adl
- openEHR-EHR-SECTION.family_history.v1.adl
- openEHR-EHR-SECTION.history_medical_surgical.v1.adl
- openEHR-EHR-SECTION.hormonetherapy.v1.adl
- openEHR-EHR-SECTION.post_radical_prostatectomy
- openEHR-EHR-SECTION.radiotherapy.v1.adl

These new 14 archetypes (four COMPOSITION and ten SECTION archetypes), together with existing COMPOSITION and SECTION archetypes, are used to organise the ENTRY and lower-level archetypes into suitable headings to ease recording. All of the newly developed COMPOSITIONs are of persistent type, denoting that the instance information is regarded as summary information rather than describing a particular event.

Finally, it is proposed to modify and/or extend four existing archetypes, to make them more generally applicable and to specialise one archetype from a generic archetype to satisfy the more detailed recording requirements. Details about the proposed extensions and modifications to existing *openEHR* archetypes are listed in Table 3.2. They are renamed to become “v2” to distinguish them from the previous version.

Archetype (Source)	Description of change(s) carried out
MODIFIED and/or EXTENDED	
CLUSTER.anatomical_location-precise.v2 (<i>openEHR</i>)	Added the value “Apical” to the list of DV_CODED_TEXT values under ELEMENT[at0006] (Aspect)
CLUSTER.specimen.v2 (<i>openEHR</i>)	Added slot of type CLUSTER[at0081] named “Secondary processing details” for CLUSTER.specimen_preparation under CLUSTER[at0000] (Specimen)
	Added ELEMENT of type DV_TEXT named “Specimen characteristics” under CLUSTER[at0000] (Specimen)
	Added ELEMENT of type DV_TEXT named “Collected at” under CLUSTER[at0050] (Collection and handling)
OBSERVATION.urinalysis.v1/v2 (<i>openEHR</i>)	Added slot of type CLUSTER[at0180] named “Specimen” for CLUSTER.specimen under ITEM_TREE[at0003] to be able to describe the specimen used in the analysis
	Added DV_CODED_TEXT choice with allowed values “Negative” and “Positive” to ELEMENT[at0095] (Protein result) to allow for the recording of the presence or absence of protein, respectively, when the result is not quantified
	Added DV_CODED_TEXT choice with allowed values “Negative” and “Positive” to ELEMENT[at0032] (Blood) to allow for the recording of the presence or absence of blood when the result is not quantified
CLUSTER.microscopy_prostate_carcinoma.v1/v2 (<i>openEHR</i>)	Added CLUSTER[at0281] (Core Details) under CLUSTER[at0160] (Tumour quantification) with the following items: - a slot of type CLUSTER[at0283] for CLUSTER.anatomical_location_precise named “Location of cores in prostate” - an ELEMENT[at0284] of type C_DV_QUANTITY named “Number of positive cores at this location” - an ELEMENT[at0285] of type DV_PROPORTION named “Proportion of positive cores at this location”
	Modified the description for Total Gleason Score for ELEMENT[at0197] (Total Gleason Score) in the archetype ontology section to “The sum of the individual Gleason scores.” instead of “The sum of the primary, secondary and tertiary Gleason scores.”
	Added CLUSTER[at0286] named “Capsular Invasion” under CLUSTER[at0000] (Microscopic findings - Prostate cancer) with an ELEMENT[at0287] named “Presence” of type DV_CODED_TEXT with possible values “Present” and “Absent”

	Added CLUSTER[at0290] named “Extra Capsular Extension” under CLUSTER[at0000] (Microscopic findings - Prostate cancer) with the following items: - an ELEMENT[at0291] of type DV_CODED_TEXT named “Presence” with possible values “Present” and “Absent” - a slot of type CLUSTER[at0294] named “Locations of extra capsular extension” for CLUSTER.anatomical_location_precise
	Added CLUSTER[at0295] called “Prostate Intraepithelial Neoplasia” under CLUSTER[at0000] (Microscopic findings - Prostate cancer) with the following items: - an ELEMENT[at0296] of type DV_CODED_TEXT named “Presence” with possible values “Present” and “Absent” - an ELEMENT[at0298] of type DV_CODED_TEXT named “Grade” with possible values “High”, “Medium” and “Low”
	Added “Perineural Involvement”, “Focal atrophy” and “Atypia” to the list of DV_CODED_TEXT values under ELEMENT[at0178] (Additional finding)

Table 3.2: Modifications carried out on existing archetypes

Several difficulties were encountered during the process of finding suitable archetypes and of establishing a direct mapping between the fields in the PCRC BIMS database to the elements in the set of existing archetypes. These are discussed in chapter 5.

Table 3.3 provides a summary of the set of archetypes needed to model the information in the PCRC BIMS.

Archetype	Source
MODIFIED & EXTENDED: openEHR-EHR-CLUSTER.anatomical_location-precise.v2.adl openEHR-EHR-OBSERVATION.urinalysis.v2.adl openEHR-EHR-CLUSTER.specimen.v2.adl openEHR-EHR-CLUSTER.microscopy_prostate_carcinoma.v2.adl	openEHR openEHR openEHR openEHR
NEW: openEHR-EHR-ACTION.radiotherapy.v1.adl openEHR-EHR-ADMIN_ENTRY.donor_admission.v1.adl openEHR-EHR-ADMIN_ENTRY.specimen.v1.adl openEHR-EHR-COMPOSITION.outcome_summary.v1.adl (persistent) openEHR-EHR-COMPOSITION.patient_background.v1.adl (persistent) openEHR-EHR-COMPOSITION.specimen.v1.adl (persistent) openEHR-EHR-COMPOSITION.treatment_summary.v1.adl (persistent) openEHR-EHR-EVALUATION.age.v1.adl openEHR-EHR-EVALUATION.psa_velocity.v1.adl openEHR-EHR-INSTRUCTION.intervention.v1.adl openEHR-EHR-SECTION.administrative.v1.adl openEHR-EHR-SECTION.chemotherapy.v1.adl	

openEHR-EHR-SECTION.current_medical_history.v1.adl	
openEHR-EHR-SECTION.current_medication.v1.adl	
openEHR-EHR-SECTION.current_symptoms.v1.adl	
openEHR-EHR-SECTION.family_history.v1.adl	
openEHR-EHR-SECTION.history_medical_surgical.v1.adl	
openEHR-EHR-SECTION.hormonetherapy.v1.adl	
openEHR-EHR-SECTION.post_radical_prostatectomy	
openEHR-EHR-SECTION.radiotherapy.v1.adl	
NO CHANGE:	
openEHR-DEMOGRAPHIC-ORGANISATION.organisation.v1.adl	<i>openEHR</i>
openEHR-DEMOGRAPHIC-PARTY_IDENTITY.organisation-name.v1.adl	<i>openEHR</i>
openEHR-EHR-ACTION.intravenous_fluid_administration.v1.adl	<i>openEHR</i>
openEHR-EHR-ACTION.medication.v1.adl	<i>openEHR</i>
openEHR-EHR-ACTION.procedure.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.citation.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.ethnic_background.v3.adl	NHS
openEHR-EHR-CLUSTER.exam-generic.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.individual_personal.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.lymph_node_metastases.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.palpation.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.physical_properties.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.specimen_preparation.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.symptom-pain.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.symptom.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.tumour_invasion.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.tumour_resection_margins.v1.adl	<i>openEHR</i>
openEHR-EHR-CLUSTER.tnm_staging-prostate.v1.adl	<i>openEHR</i>
openEHR-EHR-COMPOSITION.report.v1.adl	<i>openEHR</i>
openEHR-EHR-EVALUATION.clinical_synopsis.v1.adl	<i>openEHR</i>
openEHR-EHR-EVALUATION.excluded-intervention.v1.adl	<i>openEHR</i>
openEHR-EHR-EVALUATION.exclusion-family_history.v1.adl	<i>openEHR</i>
openEHR-EHR-EVALUATION.problem-diagnosis.v1.adl	<i>openEHR</i>
openEHR-EHR-EVALUATION.problem.v1.adl	<i>openEHR</i>
openEHR-EHR-ITEM_TREE.intravenous_fluids.v1.adl	<i>openEHR</i>
openEHR-EHR-ITEM_TREE.medication.v1.adl	<i>openEHR</i>
openEHR-EHR-OBSERVATION.exam.v1.adl	<i>openEHR</i>
openEHR-EHR-OBSERVATION.imaging.v1.adl	<i>openEHR</i>
openEHR-EHR-OBSERVATION.lab_test-histopathology.v1.adl	<i>openEHR</i>
openEHR-EHR-OBSERVATION.story.v1.adl	<i>openEHR</i>
SPECIALIZED:	
openEHR-EHR-OBSERVATION.lab_test-psa.v1.adl	<i>openEHR</i>

Table 3.3: Summary of archetypes used to cover the PCRC BIMS concepts

3.7 Design of templates for the PCRC BIMS

As explained in Section 2.2.11, archetypes are used at run-time through templates that hierarchically combine the archetypes into complete Compositions and which may be used by the EHR system to create screen forms. Templates define the tree of the particular group of archetypes that are to be used in a specific circumstance and allow further constraints on them, which can then be translated by the application level software into specific screen forms.

Thus, once the set of archetypes to cover all the information of the BIMS database was defined, these were arranged in templates that specify, combine and constrain the set of archetypes to represent the local context of the PCRC BIMS. This task corresponds to the *omda* Step 5: “Design templates”. Since archetypes are intended to be generic and provide maximal data sets, the process of creating templates is used to localise and adapt the chosen generic archetypes to the specific context at hand, thus “softening the rough” edges inherent in using a small information reference model. The Ocean Informatics Template Designer⁹ (license required) was used for template design.

While the design of these templates does not add towards the goal of enabling interoperability, they have been created for the purposes of demonstrating how the generic archetypes that cover the information in the PCRC BIMS can be adapted to the specific information needs of the PCRC BIMS. The templates constrain the complete set of options provided by the archetypes to the ones needed in the PCRC BIMS, thus representing only the fields needed by this particular biobank. The step of designing the templates using the defined archetypes completes the process of localising archetypes to the specific needs of the PCRC BIMS. The resulting templates can then be used directly to closely reproduce the screen layout of the legacy PCRC BIMS screen interface in a generic archetype-based system implementation.

During this process, the aim was to adhere to the PCRC BIMS table structure as closely as possible. In most cases, one template was created that corresponded to one table in the PCRC BIMS database, so that the application software can create one screen form for each template. Thus, the set of above archetypes was arranged into a set of 18 templates that reflect the structure of the PCRC BIMS conceptual database schema and could be used by application level software to create screen forms that correspond closely to the tables as they are currently used in the Distiller software. The screen forms created from the templates that constrain the archetypes could also be used to discuss the data structures with the biobank’s researchers to make sure that the content of the legacy database is covered completely and correctly. The new archetypes and templates and template-auxiliary files can be found on the CD provided.

⁹Ocean Informatics Template Designer, <http://www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-template-designer.html>

Table 3.4 names the templates created for the PCRC BIMS, corresponding to the tables in the PCRC BIMS databases, using the set of archetypes defined above.

TEMPLATES
Patient.oet
Biochemistry.oet
Pre-Op PSA.oet
Post-Op PSA.oet
Patient Outcome.oet
Treatment Options.oet
Outcome.oet
BPH.oet
Biopsy.oet
Radical Prostatectomy.oet
Sample Collection - Blood.oet
Plasma.oet
Serum.oet
DNA.oet
RNA.oet
Sample Collection - Tissue.oet
Fresh - RNALater.oet
Sample Collection - Urine.oet

Table 3.4: Summary of templates for the PCRC BIMS

A screenshot of the “Sample Collection - Urine.oet” template as shown in the Ocean Informatics Template Designer is shown in Figure 3.8.

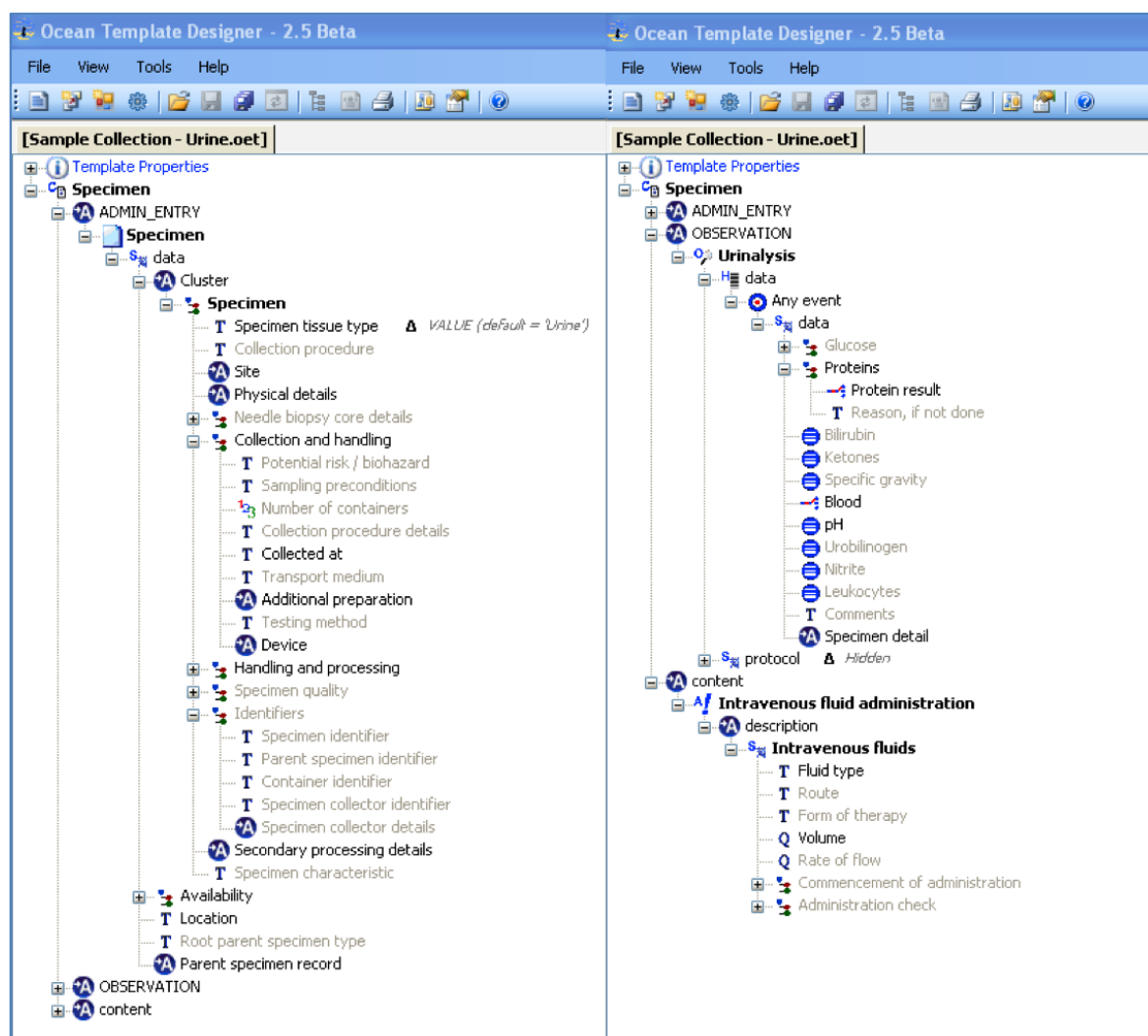


Figure 3.8: Screenshot of “Sample Collection - Urine.oet” template (left: showing ADMIN_ENTRY, other archetypes collapsed, right: ADMIN_ENTRY collapsed, showing OBSERVATION and ACTION)

A screenshot of a screen form generated using this template and the included archetypes using a prototype implementation of an *openEHR* system, the *openEHR-gen Framework*¹⁰ (see Appendix H for a discussion on *openEHR* system implementations) is shown in Figure 3.9.

¹⁰*openEHR-gen Framework*, <http://code.google.com/p/open-ehr-gen-framework/>

section.COMPOSITION-specimen.v1

section.COMPOSITION-patient_simple.v1

Specimen:

Location:

Room 3.4, blue freezer

Specimen:

Specimen tissue type:

Urine

Collection and handling:

Collected at:

Radical Prostatectomy

Handling and processing:

Date and time of collection:

29 October 2012 09:22

Date and time processed:

29 October 2012 09:22

Identifiers:

Container identifier:

1.3.4

Urinalysis:

Proteins:

Protein result:

Negative

Blood:

Positive

pH:

7.0

Intravenous fluid administration:

Fluid type:

IV fluids

Volume:

(*..*) 500 ml

Guardar

Figure 3.9: Screenshot of screen form generated from “Sample Collection - Urine.oet” template by the openehr-gen Framework

Using these templates, the fields in the PCRC database can be mapped to leaf data types in the archetypes composed in the templates. Some examples are given below. These use the ADL path syntax as defined in the *openEHR* specification for ADL [50]

For example, the field “Race (choice: Asian, Caucasian,)” in the root “Patient” table in the PCRC BIMS database can be mapped to the field with the following path in the COMPOSITION.patient_background archetype as constrained in the “Patient.oet” template:


```
[openEHR-EHR-COMPOSITION.patient_background.v1]
  /content [openEHR-EHR-SECTION.administrative.v1 and name/value='Administrative']
    /items [openEHR-EHR-ADMINENTRY.donor_admission.v1 and name/value='Donor admission'
    ]
      /data[at0001]
        /items [openEHR-EHR-CLUSTER.individual_personal.v1 and name/value='Individual's
        personal demographics']
          /items [openEHR-EHR-CLUSTER.ethnic_background.v3 and name/value='Ethnic
          Background']
            /items[at0001] (Subject ethnicity)
              /items[at0002] (Ethnicity) = 'Asian'
```

Listing 3.1: Path in “Patient.oet” template to model the patient’s ethnicity (“Race” in the PCRC BIMS table “Patient”)

To model the fields “Date Collected (Date)” and “Time of Collection (Time)” in the table “Sample Collection - Blood”, the path in the COMPOSITION as constrained by the template “Sample Collection - Blood.oet” is as follows:

```
[openEHR-EHR-COMPOSITION.specimen]
  /content [openEHR-EHR-ADMINENTRY.specimen.v1 and name/value='Specimen']
    /data[at0001]
      /items [openEHR-EHR-CLUSTER.specimen.v1 and name/value='Specimen']
        /items[at0046] (Handling and processing)
          /items[at0015] (Date and time of collection) = '2001-05-12T07:35'
```

Listing 3.2: Path in “Sample Collection - Blood.oet” template to model the time and date of blood collection

It should be noted that this path leads to a leaf data item of data type DV_DATETIME, which combines both the date and the time of the collection in one instance value. Furthermore, since the sample type, i.e. blood, is not modelled on the level of the archetype, but provided as instance data, the leaf data item at the following path in an instance of the “Blood.oet” template needs to be populated with the instance value “blood”:

```
[openEHR-EHR-COMPOSITION.specimen]
  /content [openEHR-EHR-ADMINENTRY.specimen.v1 and name/value='Specimen']
    /data[at0001]
      /items [openEHR-EHR-CLUSTER.specimen.v1 and name/value='Specimen']
        /items[at0029] (Specimen tissue type) = 'blood'
```

Listing 3.3: Path in “Sample Collection - Blood.oet” template to instantiate the sample type as blood

The “Sample Collection - Blood.oet” template can be constrained to contain blood as the default and only possible value to ease data recording, since in the template for blood, the only value that would be recorded here is ‘blood’.

Similarly, the “Urine pH” value in the “Sample Collection - Urine” table maps to leaf item at the following path in the COMPOSITION.specimen archetype as constrained in the “Sample Collection - Urine.oet” template:

```
[openEHR-EHR-COMPOSITION.specimen]
/content [openEHR-EHR-OBSERVATION.urinalysis.v1 and name/value='Urinalysis']
/data[at0001]
/events[at0002]
/data[at0003]
/items[at0126] (pH) = '7.1'
```

Listing 3.4: Path in “Sample Collection - Urine.oet” template to model the “Urine pH” in the “Sample Collection - Urine” table

The field “Chemotherapy: Drug (short text)” in the “Treatment Options” table is mapped to the “Name of medication” field in the “Treatment Options.oet” template as follows:

```
[openEHR-EHR-COMPOSITION.treatment_summary.v1]
/content [openEHR-EHR-SECTION.post_radical_prostatectomy.v1 and name/value='Post
radical prostatectomy']
/items [openEHR-EHR-SECTION.chemotherapy.v1 and name/value='Chemotherapy']
/items [openEHR-EHR-ACTION.medication.v1 and name/value='Medication action']
/description [openEHR-EHR-ITEM.TREE.medication.v1]
/items[at0001] (Name of medication) = 'Docetaxel'
```

Listing 3.5: Path in “Treatment Options.oet” template to model the “Chemotherapy: Drug (short text)” in the “Treatment Options” table

The free text family history instance data in the “Patient” table maps to the data item at the following path in the “Patient.oet” template:

```
[openEHR-EHR-COMPOSITION.patient_background.v1]
/content [openEHR-EHR-SECTION.family_history.v1 and name/value='Family history']
/items [openEHR-EHR-EVALUATION.clinical_synopsis.v1 and name/value='Clinical
synopsis']
/data[at0001]
/items[at0002] (Synopsis) = 'Father died of prostate cancer at age 57, no
other cancer in the family'
```

Listing 3.6: Path in “Patient.oet” template to model the “Family History (long text)” in the “Patient” table

Similar to the blood sample, the sample type, i.e. ‘urine’ needs to be specified as an instance value in the CLUSTER.specimen archetype that is specified in the “Sample Collection - Urine.oet” template (again, this value can be defaulted on the the template level, so that the value is automatically added when the user wants to populate the archetypes constrained within the “Sample Collection - Urine.oet” template).

As already shown in the case of date/time of collection for a blood sample, an exact one-to-one mapping could not always be established, since the information in the source model (the PCRC BIMS biobank database schema) and the target model (existing *openEHR* archetypes) is modelled differently, sometimes in incompatible ways. The problems that have been encountered and the consequences these have for using archetypes to share legacy data are discussed in Chapter 5.

3.8 Summary

This chapter described the modelling of the information encoded in a specific prostate cancer biobank, the PCRC biobank with openEHR archetypes and templates, so that the information in this biobank can be completely represented with archetypes and templates. This work presents an important step towards the goal of investigating the feasibility of using the *openEHR* approach as a candidate approach for the proposed eBMRR.

The outcome of this experiment was that it is possible to model all of the information in the PCRC database with archetypes. The approach followed in this experiment, which basically follows the same steps as proposed by the *odma* approach developed concurrently by Buck et al. [57] to model the data in the PCRC database was useful to achieve this task. Furthermore, the majority of the fields in the PCRC BIMS database, including sample-related fields, could be represented with existing archetypes, originally developed for the creation of EHRs, which could be reused without change in the biobank context, thus reducing modelling effort. This indicates that the underlying principle of reuse of archetypes is valid and that archetypes developed to represent clinical concepts in EHRs can generally be reused to represent commonly recorded data in biobanks. The current database does not provide any meta-information beyond the hierarchical relationship of the tables and the names of the tables and field, so to fully understand the data, external documentation gathered and provided by the research nurses would need to be consulted. Due to this, it is difficult to share the data in the database with collaborating institutions or use it for future (possibly integrative) research. Using archetypes and templates to record the information that is currently stored in the PCRC database allows for the data to be recorded in context, augmenting the data with the meta-data necessary to understand the data within the biobank in an globally understandable context, without the need to consult external documentation.

However, using archetypes to represent the knowledge in the PCRC biobank so that they can be used in an implementation to store the information the biobank was not without challenges. A detailed evaluation of the attempt to model the information in this biobank with *openEHR* archetypes, describing the challenge of supporting sample management and further challenges encountered and their consequences for making legacy data available through archetypes is given in Chapter 5.

To conclude, this experiment provides evidence that in the future it may be possible to extract relevant clinical data for the biobank directly from archetype-compliant EHR systems, thus reducing the effort needed in populating the biobank manually with this information from the patients' medical records.

The current PCRC database does not provide the option to record the research data

gathered through further analysis on the biospecimen collected in this biobank, so that each researcher stores the results of their research on their own computer. This data may in the worst case be lost when the researcher leaves the institution or, in the best case, it may be published in a paper and the raw data uploaded to a specific database. However, no link will be established between the data generated and the biobank record and thus future integrative data analysis is not possible. Ideally, the repository would allow for research results gathered from experiments on patient samples to be stored together with the clinical patient data in the database and the operators of the PCRC biobank have expressed the wish to be able to store this data in addition to the core data currently recorded in the database. However, this option requires the identification and definition of the data that they wish to record, which, due to the large variety of research data would be a time-consuming process. In an archetype-based record, if generic archetypes could be developed to cover the results of research, such as omic data won through high-throughput analysis on patient specimens, then this research information could be generically integrated with patient information as proposed by the creation of the eBMRR. While it may not be feasible to model the complete raw data from omic experiments through archetypes in the short-term, as a step forward, initially archetypes may be designed to model a subset of key observations and conclusions while embedding the raw data in its original format. The availability of generic archetypes for (omic) research results would reduce development time of eBMRRs in specific contexts, while making the data semantically interoperable and future-proof.

An approach for the development of archetypes to store research information, specifically from high-throughput analysis, such as omic information, is described in the next chapter.

Chapter 4

Development of ‘omic’ archetypes

4.1 Introduction

During the process of modelling the PCRC biobank database with archetypes and templates, described in Chapter 3, it was found that many existing archetypes that had been developed for the creation of EHRs can be reused in the context of the biobank to model clinical information. However, the aim of the proposed eBMRR is to augment the clinical patient information with the results from experiments carried out on the patient’s biospecimens. Thus, to do this with the *openEHR* approach, archetypes that model the results of biomedical analysis, for example from high-throughput experiments on patient samples need to be developed. Currently there are no archetypes that model this type of data. This chapter describes the development of omic archetypes to model research results from omic analysis on patient biospecimens.

To develop a set of generic omic archetypes that can be used to record the results of high-throughput analysis on patient biospecimens, the general concepts that need to be recorded need to be identified first. Then the particular data that should be recorded under each concept needs to be identified, down to the level of the data type, the name of a parameter, a description, allowed occurrences, allowed run-time values or value ranges and units. This information can then be used to design omic archetypes. The definition and structuring of complex data is one of the most difficult and costly parts of information intensive IT projects. Thus, instead of starting “from scratch”, it was thought that the comprehensive work on omic reporting standards that has been or is being carried out could be leveraged to support the development of omic archetypes. Similar to the efforts in the clinical domain to standardise the recording of clinical information to create a shareable electronic health record, omic reporting standards were motivated by the need to standardise the recording of omic information in order to facilitate the sharing of omic information. In this regard it is advantageous that all omic reporting standards are consensus-based standards produced by the community and thus already represent a community view on what information should be included, which parallels recommended archetype development processes [69, 68, 66, 236].

The following section describes how the work on omic reporting standards, specifically their accompanying data exchange formats, is exploited to facilitate the development of omic archetypes.

4.2 Leveraging omic data standards in the development of omic archetypes

As described in Chapter 2, omic data standards are generally comprised of three parts [156]:

- a minimum information (MI) specification, which defines the minimum information that should be recorded about an omic experiment in plain text, often accompanied by a simple checklist,
- a data exchange format, often based on XML, that can be used to at least encode the minimum information specified in the MI specification but generally allows for more information to be recorded, and
- an accompanying ontology/controlled vocabulary (CV) to be used to define new data elements or annotate specified data elements in the data exchange format.

Both the MI specifications and the accompanying checklists specify the minimum content that needs to be recorded to allow external researchers to make their own conclusions or repeat the experiment [20], but they do not dictate the format of *how* the data should to be recorded. MI specifications are text-based pdf documents, describing the information to be provided in prose, partially organised into hierarchical lists. An excerpt from the MIAPE-MS guidelines, the MI specification covering mass spectrometry experiments, is given in Listing 4.1. A copy of the full guidelines is provided in Appendix B.

```
Reporting guidelines for mass spectrometry
1. General features
1.1 Global descriptors
- Responsible person (or institutional role if more appropriate); provide name,
  affiliation and stable contact information
- Instrument manufacturer and model
- Customisations (summary)

2. Ion sources
As each spectrum is acquired using only one ionisation source, select the one that
applies
2.1 Electrospray Ionisation (ESI)
- Supply type (static, or fed)
- Interface manufacturer, model
- Sprayer type, manufacturer, model
```

```

- Other parameters if discriminant for the experiment

2.2 MALDI

[.....]

```

Listing 4.1: Excerpt from reporting guidelines for mass spectrometry from MIAPE-MS v2.97.8

The MI checklists, which have been created for most MI specifications, provide some more detail about what should be recorded and in a more structured format, organised by index number, item name, item value, definition and examples, although only index number and item name are consistently provided, with the other fields often left empty.

MI checklists can be viewed and downloaded in a number of different formats, including HTML table, XML schema, tab-delimited files from the MICheckout tool on the MIBBI website¹. An excerpt from the MIAPE-MS checklist, downloaded in the tab-delimited format is given in Listing 4.2.

```

<!-- MIBBI\_Foundry\_compilation 2012-06-20T11.09.35+01.00 -->
Index no. Item name Item value Definition Examples
1. Mass\_spectrometry
1.1. General features
1.1.1. Global descriptors
1.1.1.1. Date
1.1.1.1.1. Date given as YYYY-MM-DD
1.1.1.2. Person or responsible role
1.1.1.2.1. Name Name should either be of the person or the institutional role
holding responsibility for the associated (meta)data.
1.1.1.2.2. Affiliation
1.1.1.2.3. Stable contact information Such information should be one or more from
the following list: email; telephone (plus international dialling code); postal
address.
1.1.1.2.4. Globally-unique person ID (authority and identifier) Where available,
please give a unique person identifier and the system of which it is part (e.g.,
ISNI, ResearcherID, etc.).
1.1.1.3. Instrument manufacturer and model
1.1.1.4. Customisations Summary thereof.
1.1.2. Control and analysis software
1.1.2.1. Software name and version
1.1.2.2. Switching criteria Tandem instruments only.
1.1.2.3. Isolation width Global, or by MS level
1.1.2.4. Location of parameters file
1.2. Ion sources As each spectrum is acquired using only one ionisation source,
select only the subcategory that applies.
1.2.1. Electrospray ionisation
1.2.1.1. Supply type Static, fed.
1.2.1.2. Interface manufacturer model and catalog number where available
1.2.1.3. Sprayer type, coating, manufacturer, model and catalogue number where
available
1.2.1.4. Relevant voltages where appropriate Provide tip, cone, acceleration

```

¹MICheckout, <http://mibbi.sourceforge.net/MICheckout.htm>

1.2.1.5. Other parameters if discriminant for the experiment	Nebulising gas ,
pressure .	
1.2.2. MALDI	
[.....]	

Listing 4.2: MIAPE-MS checklist in tab-delimited format downloaded from MICheckout

Both guidelines and checklists determine what information needs to be minimally provided in a high-level manner, without providing much detail on the exact nature of the information, e.g. “Description of the ion source and relevant parameters” or “Parameters used in the generation of peak lists or processed spectra” or “Ion mode”, without going into detail about what these parameters may be or what values they could take. They do not specify the details on how this information should be recorded, such as defining the data type, for example, whether to use a text or a numeric value to describe a certain piece of information, nor do they provide an indication of which units to use. For example, in the MIAPE-MS checklist, point 1.2.1.4. only specifies to record “Relevant voltages where appropriate” and further to “Provide tip, cone, acceleration”. Both MI specifications and checklists are meant to be used with ontologies, such as the PSI-MS ontology for mass spectrometry for the MIAPE-MS guidelines, but no details about the usage of particular ontology terms are provided.

There are many alternative ways of recording data that comply with MI specifications. Thus, compliance with MI specifications alone does not ensure interoperability [135]. Without detailed specifications on how the data should be recorded, such as specifying the exact parameter, data type, the name of a parameter, a description, allowed occurrences, allowed run-time values or value ranges and units for each piece of information, interoperability cannot be guaranteed [135]. Realising this fact, many omic standard development organisations have developed/are developing accompanying omic data exchange formats that encode the information specified in MI guidelines and that can be used as common data formats to support the exchange of omic data.

These data exchange formats or markup languages can be used to create instance files that comply with their parent MI specification which they encode, but they generally also allow the creation of files that contain more or less than the information specified in the MI specification. In general, the defined data exchange formats allow a much more comprehensive set of information to be recorded. This aligns well with the concept of archetypes, since archetypes aim to cover the maximum information about a concept [237, 17], rather than the minimum information only. As an example of a markup language to exchange omic data, an extract of mzML, the markup language developed by HUPI PSI which can be used to record the results from mass spectrometry experiments, is shown in Listing 4.3 in XML schema format.


```

<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://psi.hupo.org/ms/mzml" version="1.1.0">
<xs:complexType name="mzMLType">
  <xs:annotation>
    <xs:documentation>This is the root element for the Proteomics Standards Initiative
      (PSI) mzML schema, which is intended to capture the use of a mass
      spectrometer, the data generated, and the initial processing of that data (to
      the level of the peak list).
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="cvList" type="dx:CVListType" />
    <xs:element name="fileDescription" type="dx:FileDescriptionType" />
    <xs:element minOccurs="0" name="referenceableParamGroupList" type="
      dx:ReferenceableParamGroupListType" />
    <xs:element minOccurs="0" name="sampleList" type="dx:SampleListType" />
    <xs:element name="softwareList" type="dx:SoftwareListType" />
    <xs:element minOccurs="0" name="scanSettingsList" type="dx:ScanSettingsListType" />
    <xs:element name="instrumentConfigurationList" type="
      dx:InstrumentConfigurationListType" />
    <xs:element name="dataProcessingList" type="dx:DataProcessingListType" />
    <xs:element name="run" type="dx:RunType" />
  </xs:sequence>
  <xs:attribute name="accession" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation>An optional accession number for the mzML document used for
        storage, e.g. in PRIDE.
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="version" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation>The version of this mzML document.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="id" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation>An optional id for the mzML document used for referencing from
        external files. It is recommended to use LSIDs when possible.</
        xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="FileDescriptionType">
  <xs:annotation>
    <xs:documentation>Information pertaining to the entire mzML file (i.e. not
      specific to any part of the data set) is stored here.</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="fileContent" type="dx:ParamGroupType">
      <xs:annotation>
        <xs:documentation>This summarizes the different types of spectra that can be
          expected in the file. This is expected to aid processing software in
          skipping files that do not contain appropriate spectrum types for it. It
          should also describe the nativeID format used in the file by referring to
          an appropriate CV term.</xs:documentation>
        </xs:annotation>

```

```

</xs:element>
<xs:element minOccurs="0" name="sourceFileList" type="dx:SourceFileListType"/>
<xs:element minOccurs="0" maxOccurs="unbounded" name="contact" type="
    dx:ParamGroupType"/>
</xs:sequence>
</xs:complexType>
.....
</mzML>

```

Listing 4.3: Extract from mzML schema

Like MI specifications, data exchange formats may be associated with controlled vocabularies (CVs) or ontologies for the annotation of the data elements in the data exchange formats. An excerpt from the ontology to be used in conjunction with MIAPE-MS is shown in Listing 4.4 (in obo format²).

```

[.....]

[Term]
id: MS:1000016
name: scan start time
def: "The time that an analyzer started a scan, relative to the start of the MS run."
    [PSI:MS]
xref: value-type:xsd\:float "The allowed value-type for this CV term."
is_a: MS:1000503 ! scan attribute
relationship: has_units UO:0000010 ! second
relationship: has_units UO:0000031 ! minute

[Term]
id: MS:1000017
name: Scan Function
def: "Describes the type of mass analysis being performed. Two primary modes are:
    typical acquisition over a range of masses (Mass Scan), and Selected Ion Detection
    . The primary difference is that Selected Ion Detection produces a single value
    for the signal at the selected mass rather than producing a mass spectrum." [
    PSI:MS]
relationship: part_of MS:1000479 ! purgatory

[Term]
id: MS:1000018
name: scan direction
def: "Direction in terms of m/z of the scan for scanning analyzers (low to high, or
    high to low)." [PSI:MS]
relationship: part_of MS:1000441 ! scan

[.....]

```

Listing 4.4: Excerpt from the PSI-MS ontology in obo format

In addition to specifying the particular ontology that should be used in conjunction with a particular data format, some data formats come with an accompanying XML-based mapping file that specifies the rules for the usage of the ontology terms for annotating an instance of an element in the data exchange file. An excerpt of this

²The OBO Flat File Format Specification, <http://www.geneontology.org/GO.format.obo-1.2.shtml>

mapping file is shown in Listing 4.5.

```
[.....]
<!-- run -->
<CvMappingRule id="run_may" cvElementPath="/mzML/run/cvParam/@accession"
  requirementLevel="MAY" scopePath="/mzML/run" cvTermsCombinationLogic="OR">
  <CvTerm termAccession="MS:1000857" useTerm="false" termName="run attribute"
    isRepeatable="false" allowChildren="true" cvIdentifierRef="MS"/>
</CvMappingRule>
<!-- file content -->
<CvMappingRule id="filecontent_must" cvElementPath="/mzML/fileDescription/fileContent/
  cvParam/@accession" requirementLevel="MUST" scopePath="/mzML/fileDescription/
  fileContent" cvTermsCombinationLogic="AND">
  <CvTerm termAccession="MS:1000524" useTermName="false" useTerm="false" termName="
    data file content" isRepeatable="true" allowChildren="true" cvIdentifierRef="MS"
  />
</CvMappingRule>
[.....]
```

Listing 4.5: Snapshot from mapping rule file for mzML data format

These data formats, together with the ontologies and ontology mapping files define the information to be recorded in detail. They define each data element to be recorded, how often it is allowed to appear, the hierarchy of where it is allowed to appear, provide a description of the data to be recorded, the data type that may be used to record this information, and possible data values, e.g. a list of values that may be used in the instance file. A lot of the work normally required for the development of archetypes, such as deciding on an appropriate data type, how many instances of this data are allowed, allowed run-time values, how this data relates to other data, providing a description etc. can be drawn from these data formats, in conjunction with the ontologies and mapping files. Taking these points into account, it was decided that the data exchange formats combined with their accompanying ontologies could be leveraged to support the creation of omic archetypes. Using these data formats as basis for the development of omic data formats removes the need for this work to be done from scratch and thus significantly reduces the development effort for high-quality omic archetypes.

Comparing the omic data format with the *openEHR* archetype model, there are structural and semantic similarities between the omic data format and an archetype definition, which can be leveraged for the creation of omic archetypes from these data formats. An XML schema file defines the valid structure of an XML instance document, i.e. the configuration of XML elements and data types in an instance XML file that conforms with the XML schema, including constraints on where the specified elements are allowed to occur in the instance file, how many times, the allowed data type and other constraints, such as limit values or certain string patterns. Similarly, archetypes define the structure of a valid instance file that complies with the archetype definition, by defining constraints on the *openEHR* reference information model classes, with

regard to the types used, their order and location, allowed occurrences and limits on data values etc. These constraints specify what configurations of reference model class instances are considered to comply with the archetype.

Both archetypes and data formats contain formal descriptions of how information should be stored, specifying the order, description, grouping and allowed-runtime occurrences of a number of pre-defined data types. Both formats are computable. Both XML schemas and archetype definitions define information hierarchies, similarly, XML instance files and instances of archetyped data are hierarchically organised. The biggest difference between omic data formats and archetypes is that archetypes model knowledge: an archetype, by definition, has to fulfil the semantic requirement to represent a distinct domain concept, whereas the omic data formats do not need to adhere to this concept. In addition, archetypes are based on a detailed and complex underlying reference model, specifically designed to support the recording of “service and administrative events relating to a subject of care” [39]. For example, there are different classes of ENTRY archetypes, each to be used for different types of concept (for example the OBSERVATION class is used to describe observations on a subject of care). This means that although both archetypes and data formats describe how certain pieces of information should be stored, the design of archetypes is controlled by semantics, with semantic restrictions on how instance data belonging to a particular concept should be organised and recorded.

However, it was thought that a mapping between the two types of meta-information, i.e. the respective data types and their constraints on their run-time usage (such as allowed run-time occurrences and location in hierarchy) could be found. This mapping could then be used to transform the information described in the XSD file and associated CV(s) into (an) ADL file(s), allowing for at least a syntactic conversion of the omic data format and associated files into an archetype format, such as ADL. The fact that the omic data formats already provide computable descriptions of the data to be recorded, makes this process amenable to automation, thus cutting out the manual work of inputting this information into an *openEHR* archetype editor or a text editor to create ADL files and further reducing development time of omic archetypes. Thus, the structural and organisational similarities between omic data formats and archetypes is exploited in creation of archetypes to reduce overall development effort. This syntactic conversion would result in a (number of) ADL file(s) that could then be used as candidate(s) for the development of semantically sound archetypes that represent distinct domain concepts.

Another advantage of this strategy is that once a suitable mapping has been found and the process automated, the automation software may also be applied to other omic data formats, further reducing work and time needed for the development of omic

archetypes. The proposed strategy for how the omic data exchange formats and their accompanying ontologies can be used in the creation of omic archetypes is described in detail in the next section.

4.3 Creating archetypes from omic data exchange formats

As explained above, if a mapping could be established between the elements of the XML-based data format and the *openEHR* reference model, then the structural and organizational similarity between the two formats could be exploited to automatically create ADL files from the omic data formats. The content of the automatically created ADL files would then need to be reviewed and adjusted so that the content of each archetype represents a distinct domain concept, resulting in a set of generically applicable and reusable omic archetypes. The following 2-phase strategy for creating omic archetypes based on existing XML-based data exchange formats and associated CVs is proposed: In Phase 1, a syntactical transformation is carried out, where the information described in the data format in XML syntax is converted into ADL syntax to move the information in the data format into the archetype computational domain. In Phase 2, the ADL files created in Phase 1 are reviewed and reorganised according to the semantic requirements of archetype design resulting in a set of semantically sound omic archetypes.

4.3.1 Phase 1: Syntactic transformation

In Phase 1, the aim is to achieve a syntactic transformation of an XML schema into (an) ADL file(s). For this, equivalences between elements of the XML schema and associated files (CV Mappings and CVs) and the *openEHR* reference model classes and their attributes need to be established. The results of these mappings are visualised in terms of ADL in the remainder of this thesis - the abstract syntax for textually expressing archetypes, which is described in detail in the corresponding *openEHR* specification [50]. Both the omic data format and *openEHR* archetypes make use of predefined data types (the information building blocks) and specify a valid combination and allowed use of these information building blocks, for which instance files can be created that comply with the format descriptions.

So to achieve a syntactic transformation from the omic data format to candidate archetype(s), a mapping between the element and attribute definitions in an XML schema to the object and attribute constraints in an archetype definition need to be performed. This will lead to the preservation of the hierarchy, order and other restric-

tions on the use of elements as specified in the XML file in the archetype definition. In this process, the data types in the original omic data formats originate from two sources: the XML schema or the mapped CV. It is important to note at this point that the aim is not to replicate the actual structure of the XSD file itself in the resulting ADL file(s), but of the hierarchy that it specifies for an XML instance file that complies with the XML schema.

Further, the term definitions for the data items recreated in the ADL file can be extracted from the naming and descriptions of instances of elements in the XML schema or CV. Thus, most (if not all) the mandatory information needed to create syntactically valid ADL files can be extracted from the omic data formats. Once this mapping is established between the XML schema and the *openEHR* archetype model, the complete hierarchical structure of a potential XML file defined by the XSD file with associated mapping rules and CVs can be reconstructed entirely into a corresponding structure in (an) ADL file(s), with the constraints on elements defined in the XSD file corresponding to constraints on *openEHR* reference model objects in the created ADL file(s).

While one XSD file and mapped ontology terms can thus be transformed into one large ADL file, it was thought that the resulting ADL file would be too large and cumbersome to deal with as input into Phase 2. In addition, it is expected that many different concepts are encoded within the omic data format, each of which should ideally be represented by a separate, generic archetype.

Therefore, instead of creating one large ADL equivalent of the hierarchy defined in the XML data format and mapped ontologies, it was thought that it would be more manageable to create a number of smaller ADL files, each covering a different part of the data format. The data formats are generally organised such that related information is to be recorded together, by specifying it under the same element definition. Assuming that the information defined under the same element can be categorised as belonging to the same concept, this grouping is likely to be preserved within the final syntactically and semantically valid archetypes. For example, when the original data format specifies certain information to be recorded together, such as device parameters, these are likely to be part of the design of an archetype capturing device details. Furthermore, the hierarchical data relationships in the omic data format are likely to be preserved in an archetype hierarchy. Thus, the information defined in the large omic XML schema file first needs to be broken down into suitable smaller subunits that roughly correspond to an archetype concept each. Then the XML properties of those subunits need to be mapped to the various properties of an archetype. This mapping process includes several distinct tasks, which are listed briefly below:

- Defining the level of granularity of the ADL files to be created from the XML

schema file

- Mapping XSD elements and data types to appropriate *openEHR* elements and data types
- Mapping of value-type ontology terms to appropriate *openEHR* elements and data types
- Mapping XSD internal data types to archetype slots
- Mapping `xs:complexType` element and attribute definitions to archetype objects
- Mapping naming and descriptions of XSD elements and attributes to archetype ontology
- Mapping XML occurrence and use attributes to archetype object node constraints
- Representing mapped `cvParam` terms in archetypes
- Mapping `CVMappingRule` and `CVTerm` attributes to archetype object constraints
- Mapping units to *openEHR* terminology

Each of these tasks and their outcome is described in detail in Section 4.4.2 on a worked example, using the *mzML* data format for mass spectrometry as starting point.

The expected outcome of this mapping and subsequent transformation would be a set of ADL files, each mirroring the structural organisation of data in an XML instance file based on the XML schema, and each covering a different part of the data format. However, these ADL files cannot be expected to be semantically valid archetypes due to differences between the design philosophies and technologies of the original data format and archetypes. Thus, in Phase 2, each created ADL file is reviewed and modified iteratively until it fulfils the semantic requirements put on archetypes. These tasks and their solutions will be discussed in the next section.

4.3.2 Phase 2: Semantic review

As explained above, due to the differences in the philosophies and technologies used in the creation of omic data formats in comparison to the creation of archetypes, it is expected that the ADL files that have been created automatically from the omic data formats in Phase 1 may not automatically represent well-designed archetypes, i.e. distinct domain-level concepts. Specifically, archetypes have to satisfy a more comprehensive set of requirements than omic data formats, mainly because archetypes are based on a complex and large underlying reference model.

While quality criteria and quality assurance for archetypes are poorly defined, some guidance is given by the “Editorial checklist for clinical content review of archetypes”³. This checklist concentrates on the content of the candidate archetypes, i.e. on their semantic correctness, rather than on correct syntax. While this checklist was developed for clinical content, it can be applied just as well to omic content. For example, one of the main criteria for archetypes is that an archetype shall describe a single clinical concept without overlapping other archetypes, based on consensus [44], [38]. Adapted to omic content, an archetype should describe a single “omic” concept without overlapping other archetypes, based on consensus.

The ADL files created in Phase 1 may have been too granular, so that information belonging to one concept may be spread over several ADL files that have been automatically created from it. These ADL files may need to be combined into a bigger archetype or absorbed into a higher-level archetype if the information they describe belongs to the same or parent concept, respectively.

Since the omic archetypes developed in this thesis do not belong to the clinical domain in which archetype development has been long ongoing, but are specific to the omic research domain, and are thus the first of this domain, little overlap was expected to be found for these. However, to avoid and resolve overlap, developed archetypes need to be checked against existing archetypes in the *openEHR* Clinical Knowledge Manager during Phase 2 of the omic archetype development process. Hence, in Phase 2, each ADL file is reviewed individually and reorganised if necessary to satisfy archetype quality criteria and result in a reusable set of non-overlapping, generic omic archetypes. The main tasks that need to be carried during the review process include:

1. Resolve scope issues amongst candidate archetypes (structural modifications)
2. Resolve overlaps with existing archetypes (structural modifications)
3. Review internal archetype content (non-structural modifications)

The execution of this review process are described in detail in Section 4.4.5 on the worked example of candidate archetypes created in Phase 1 from the mass spectrometry data format.

The proposed 2-phase strategy is somewhat reminiscent of the two-step process that *openEHR* proposes as a technical solution to connect legacy data sources from existing systems to *openEHR* to achieve information integration. To do so, they propose first a syntactic transformation of legacy data from its original format into a format satisfying the logical structure and semantics of so-called ‘integration’ archetypes. These

³Editorial checklist, <http://www.openEHR.org/wiki/display/healthmod/Editorial+checklist+for+clinical+content+review+of+archetypes>

‘integration’ archetypes are based on a special *openEHR* reference model class called `GENERIC_ENTRY`, which is a sibling of `SECTION` and `ENTRY`, and holds completely generic, archetypable structures. This transformation carries the data into the *openEHR* computational context. In the second step, a semantic transformation is carried out on the intermediate *openEHR* data into data that represent a) instances of the main *openEHR* reference model, and b) comply with ‘designed’ clinical archetypes, using semantic transformation rules [238]. Using this process, legacy instance data can be converted into instance data complying with the *openEHR* reference model. This process expects the existence of integration archetypes and of properly ‘designed’ archetypes to which the integration archetypes can be mapped to for data conversion purposes.

Figure 4.1 depicts the 2-phase process for creating omic archetypes.

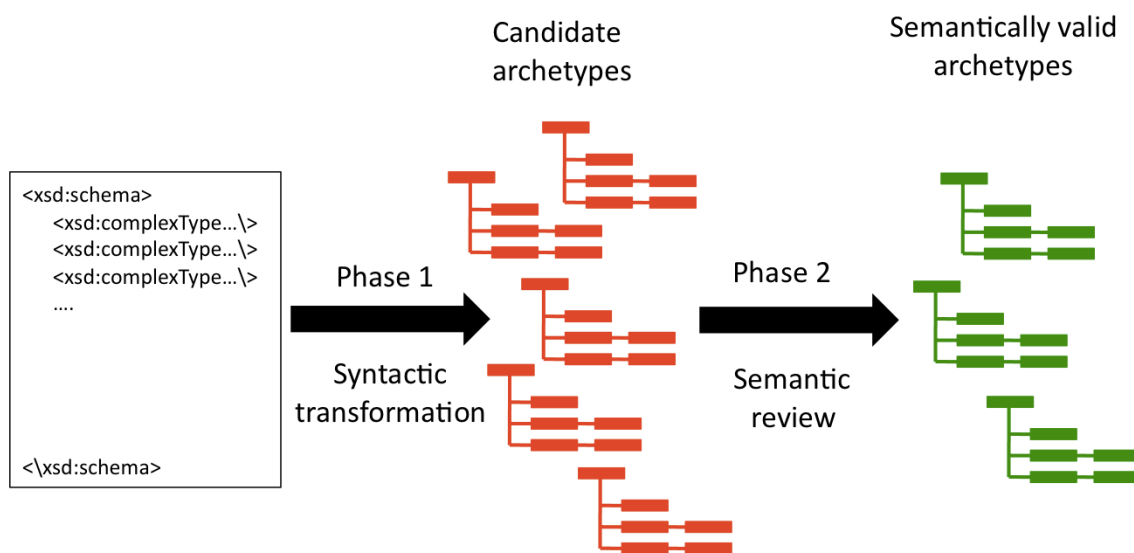


Figure 4.1: 2-phase process for creating omic archetypes

Thus, one could have suggested creating an ADL file complying with `GENERIC_ENTRY` for the complete instance structure of a maximal XML file complying with mzML, which is similar to creating one large `CLUSTER` type ADL file. Since both outcomes represent large, unwieldy files that may contain several concepts, it was decided that creating one large `GENERIC_ENTRY` type candidate archetype would not add any value over the suggested fine-grained approach of creating one `CLUSTER` type candidate archetype for each `xs:complexType` element defined in the mzML file, which are easier to deal with in the review process.

In the case of this research, in the first phase a syntactic transformation of the legacy data recording specification (the data format instead of the instance data that complies with the data format) into an *openEHR* format (ADL) is used to bring the

legacy data recording specification into the computational context of *openEHR*. The ADL files created are comparable to the ‘integration’ archetypes mentioned above. However, since no omic archetypes yet exist, these ‘integration’-type archetypes are then reviewed and modified to produce a set of semantically valid ‘properly designed’ omic archetypes. In fact, a by-product of this research is the establishment of a mapping between data in the legacy omic data formats to the properly designed omic archetypes that result from this research.

4.4 Development of omic archetypes for mass spectrometry

Mass spectrometry is a commonly used technique in the PCRC biobank to support the identification of prostate cancer biomarkers. Thus, it was decided to develop omic archetypes for the application and results of a mass spectrometry experiment, based on the mzML data format. mzML is a standard-based omic data exchange format for mass spectrometry developed by the HUPO PSI Mass Spectrometry working group (PSI-MS WG)⁴ to solve integration problems due to the existence of a number of different proprietary formats [239]. mzML is a widely used markup language for exchanging and storing the results of mass spectrometry experiments. The mzML data format was chosen as the first example to work with, because it is stable, formally specified, mature, and defined in a computable, XML-based data format, which makes it amenable to automation. To help understand the tasks involved in the mapping process to support the automatic creation of ADL files from an XML-based omic data exchange format, the following section describes the structure and characteristics of the mzML data exchange format and auxiliary files used to support the exchange of mass spectrometry data.

4.4.1 The mzML data exchange format

The mzML data exchange format is a data exchange format for recording the results of a mass spectrometry experiment in proteomics. In particular, as stated in the description of the root element of the data format, it is “intended to capture the use of a mass spectrometer, the data generated, and the initial processing of that data (to the level of the peak list)”. Mass spectrometry is a widely used technique in scientific research to investigate bio-molecules by measuring the mass-to-charge ratios of their ionised forms or fragments of them [239]. The resulting mass spectra are then further processed computationally to identify, characterise or quantify the analysed molecules.

⁴PSI-MS working group, <http://psidev.info/groups/mass-spectrometry>

A diagram describing a mass spectrometer is shown in Figure 4.2 (adapted from <http://pubs.usgs.gov/of/2001/ofr01-257/images/figure1.gif>). A mass spectrum visualising the output of the mass spectrometry experiment is presented in Figure 4.3 (from <http://www.peptracker.com/moreInformation/>).

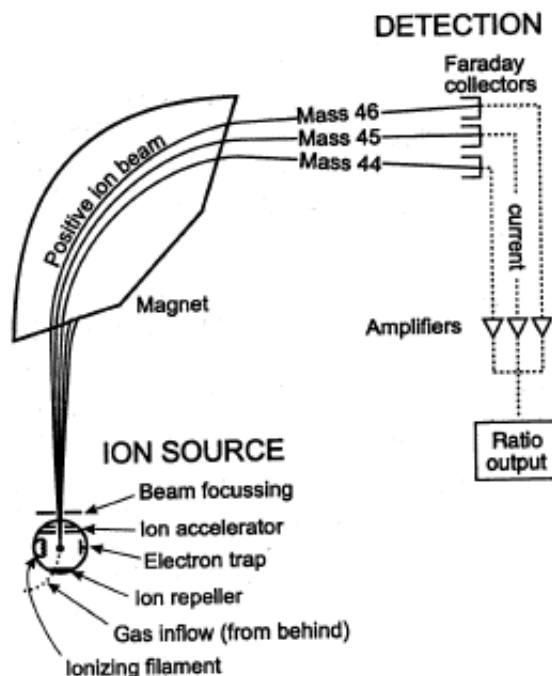


Figure 4.2: Mass spectrometer, Source: <http://pubs.usgs.gov/of/2001/ofr01-257/images/figure1.gif>

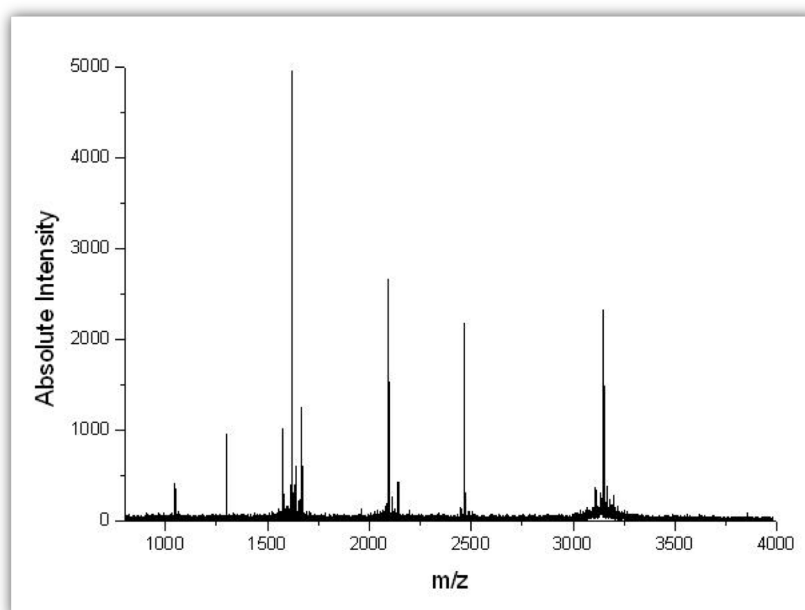


Figure 4.3: Mass spectrum, Source: <http://www.peptracker.com/moreInformation/>

The results are sensitive to differences in the various MS instruments, experimental conditions and software setup and thus experimental details need to be provided together with the results. As a consequence, the mzML data exchange format models both the actual data created, including associated metadata and a description of the instrument, configuration settings and any preprocessing steps that were applied. In terms of an experimental pipeline, mzML captures the data describing the process and results after protein separation has been carried out (protein separation is covered by the spML data exchange format) and before informatics analysis is carried out on the resultant mass spectra (the results of informatics analysis on mass spectra are covered by the proteomics informatics data exchange formats mzIdentML and mzQuantML). The design of the mzML format is based on the MIAPE-MS specification, the mass spectrometry specific module of the MIAPE guidelines, and can be used to encode MIAPE-MS requirements. mzML does not enforce compliance with MIAPE-MS, and mzML instance files can be created that contain less than the information required by MIAPE-MS [239]. mzML itself is specified as an XML schema file, defining how to create a compliant XML instance file. Most of the documentation and files and material used as basis for the development of archetypes have been sourced from the HUPO PSI development website⁵. mzML was initially released in June 2008 (version 1.0.0.0) and went through a number of revisions, the latest was been released in June 2009 (now at version 1.1.0.1). It has been stable since, with no further releases planned. As such, the mzML format is a very stable format and has been widely implemented in tools and software⁶. Several auxiliary files and supplementary information are provided together with the textual mzML specification, all of which can be found on the HUPO PSI website⁷, including associated schema definition files, documentation files, links to validation tools, a selection of sample files generated by several different kinds of software and manually and links to other relevant and useful websites, e.g. to a website for an ontology editor.

The structure of mzML

A visualization of the high-level structure of the mzML schema is shown in Figure 4.4 (adapted from [239]):

The mzML schema (mzML1.1.0.xsd) specifies a number of `xs:complexType` elements, of which the higher-level ones are represented by the coloured and named rectangles that are grouped and linked hierarchically as visualised in the diagram above. Several of the defined `xs:complexType` elements were reused in different locations in

⁵HUPO PSI, <http://www.psidev.info>

⁶List of products supporting mzML provided on <http://www.psidev.info/mzml>

⁷mzML, <http://www.psidev.info/mzml>

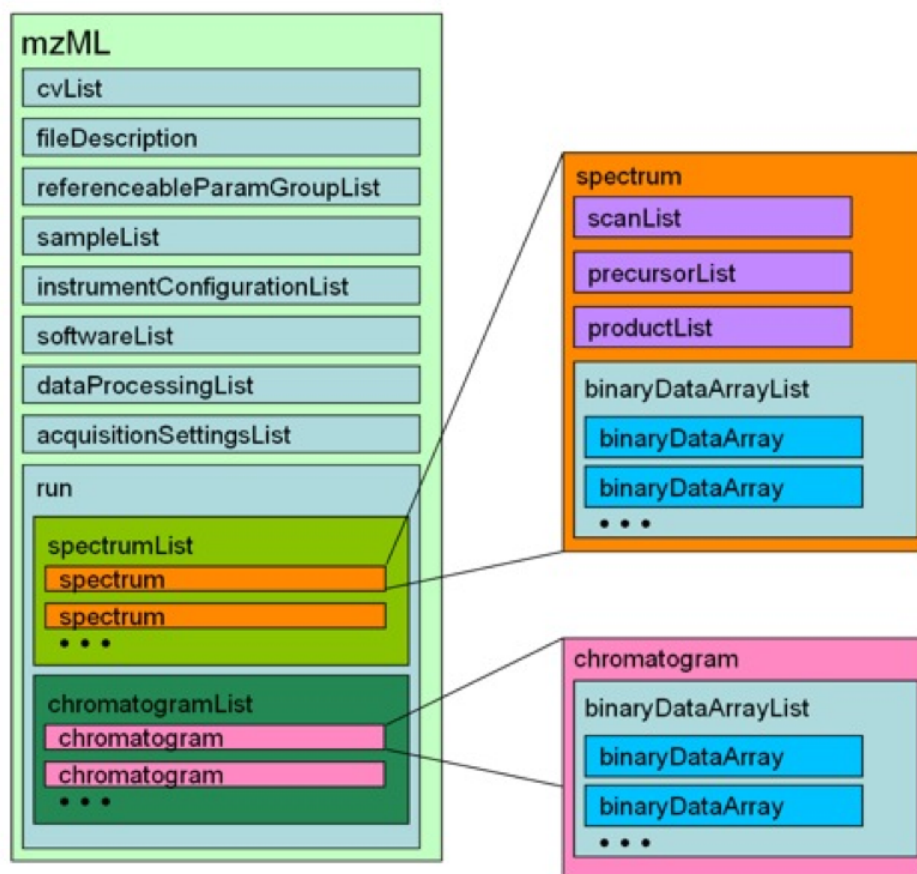


Figure 4.4: Structure of mzML, Source: [239]

the hierarchy, such `binaryDataArrayList` etc. The `mzML` element builds the root element of the XML hierarchy in an instance file complying with this schema. This root element contains several list type elements that are used to list items, e.g. `SampleList` will list Sample items. It also contains a “run” element that contains a hierarchical element definition that can be used to capture “a single, consecutive and coherent set of scans on an instrument”. Specifically, instance items that have been listed under the list types are referred to from within the run data hierarchy, thus providing context to these items.

The PSI has also developed a CV (called PSI-MS) to be used in conjunction with the `mzML` file, to annotate specific data elements [239]. An XML-based mapping file, `ms-mapping.xml`⁸ specifies in which location in an `mzML` instance file any given terms are allowed or recommended to be used [239]. The format of a `CvMappingRule` file is shown in Figure 4.5.

A mapping file contains mapping rules (`CvMappingRule` in the mapping file) whose attributes specify the allowed location of the term in the XML file (`cvElementPath`)

⁸MS mapping, <https://psidev.svn.sourceforge.net/svnroot/psidev/psi/mzml/validator/ms-mapping.xml>

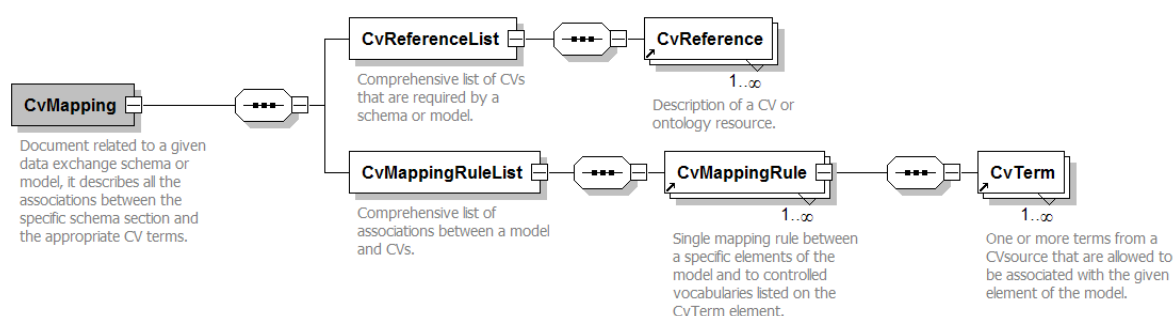


Figure 4.5: Format of the CvMappingRule file, Source: <http://www.psidev.info/validator-tutorial-3>

using XPath, the level of requirement (requirementLevel) which is either “MAY”, “SHOULD” or “MUST”, the element scope in the schema within the non-repeatable condition applies (scopePath) and how terms can be combined using “AND”/“OR”/“XOR” logic (cvTermsCombinationLogic). Then, the term to be used itself is specified (CvTerm), including the name (termName), accession number in the CV file (termAccession), an identifier for the CV used (cvIdentifierRef), and other information, relating to term validation, allowed occurrences, repeatability and use of related child terms.

A detailed explanation of the mapping file and its associated XSD file is given on the PSIDEV website⁹. The PSIDEV also provides an explanation of the elements for the mapping file (CvMapping.xsd)¹⁰.

An excerpt from the ms-mapping file is provided in Listing 4.6.

```

<CvMapping >
< >
  <!-- detector -->
  <CvMappingRule
    id="detector_must"
    cvElementPath="/mzML/instrumentConfigurationList/instrumentConfiguration/
      componentList/detector/cvParam/@accession"
    requirementLevel="MUST"
    scopePath="/mzML/instrumentConfigurationList/instrumentConfiguration/componentList
      /detector"
    cvTermsCombinationLogic="AND">
    <CvTerm
      termAccession="MS:1000026"
      useTerm="true"
      termName="detector type"
      isRepeatable="false"
      allowChildren="true"
      cvIdentifierRef="MS" />
    </CvMappingRule>
  < >
</CvMapping>
  
```

Listing 4.6: Excerpt from ms-mapping.xml

⁹<http://www.psidev.info/validator-tutorial-3>

¹⁰<http://www.psidev.info/sites/default/files/CvMapping.xsd>

In this example, a mapping rule is defined for an element in the instance mzML file at the path given by the cvElementPath attribute value (“/mzML/instrumentConfigurationList/instrumentConfiguration/componentList/detector/cvParam/@accession”). The occurrence of the specified CV term is mandatory in the XML instance file as indicated by the ”MUST” value of the requirementLevel attribute. This mapping rule applies to the scope in the instance XML file given in the scopePath attribute (“mzML/instrumentConfigurationList/instrumentConfiguration/componentList/detector”). Further, the term may be used in combination with other terms associated with this mapping rule (as specified by the “AND” value of the cvTermsCombinationLogic attribute). The term that may be used at the specified location is called “detector type” as denoted by the termName attribute, and has the accession number “MS:1000026” (according to the termAccession attribute) in the MS CV as specified in the cvIdentifierRef attribute (which refers to the MS CV listed at the start of the mapping file). Further, the useTerm attribute is set to true, which means that the term itself can be used for annotation. Further, the allowChildren attribute is set to true, which means that related child terms can also be used for annotation in addition to the term itself. However, only one instance of this term (or its childterms) is allowed at this location as specified by the isRepeatable attribute.

The code snippet in Listing 4.7 shows how the above CV mapping rule is applied in an XML instance file by including the specified CV term at the specified location, where the cvParam provides the term “electron multiplier” with the accession number “MS:1000253”, which is a child term of the ”detector type” CV term to fulfil the mandatory occurrence requirement of the mapping rule with id=“detector_must” described above.

```

<mzML ...>
  <instrumentConfigurationList .....>
    <instrumentConfiguration .....>
      <componentList .....>
        <detector .....>
          <cvParam cvRef="MS" accession="MS:1000253" name="electron multiplier" value=
            "" />
        </detector>
      </componentList>
    </instrumentConfiguration>
  </instrumentConfigurationList>
<....>
</mzML>

```

Listing 4.7: Example of a mapping file applied to an element in an mzML instance file

The next section describes how the mappings between the element descriptions of the mzML and related files and the *openEHR* reference model were established, so that the data recording information specified in the mzML file could be used to construct structurally equivalent ADL files.

4.4.2 Mapping of mzML structures to archetype structures

To prepare for the syntactic transformation of omic data exchange formats into structurally equivalent ADL files that syntactically comply with the *openEHR* reference model, a mapping between the constructs in both formats needs to be carried out. Then, a syntactic transformation can be carried out, applying the mapping in the process, thereby bringing the information of the omic data exchange formats from the XML realm into the *openEHR* realm.

This process can be broken down into a number of steps and involves several tasks, including:

- Define the level of granularity of the ADL files to be created from the XML schema file
- Map XSD elements and data types to appropriate *openEHR* elements and data types
- Mapping of value-type ontology terms to appropriate *openEHR* elements and data types
- Map XSD internal data types to archetype slots
- Map `xs:complexType` element and attribute definitions to archetype objects
- Map naming and descriptions of XSD elements and attributes to archetype ontology
- Map XML ‘occurrence’ and ‘use’ attributes to archetype object node constraints
- Represent mapped `cvParam` terms in archetypes
- Map `CVMappingRule` and `CVTerm` attributes to archetype object constraints
- Map units to *openEHR* terminology

These steps will be explained in detail in the following sections. Once a complete mapping was established for all relevant source and target elements, transformation software to carry out an automatic transformation of XML schemas into candidate archetypes was developed.

Level of Granularity

Since the desired output of the 2-phase omic archetype development process is a set of reusable omic archetypes, one decision that needed to be taken during the transformation process concerned the granularity of the initial ADL output file(s) that should be created from the omic XML schema file.

As discussed above, applying only the syntactic mapping, the information specified in the complete XML schema and associated files could be transformed into one large ADL file, mirroring the complete hierarchical structure of the XML file as defined by the XSD file and associated CVs in the ADL definition. However, the size of this ADL file would be very large and unwieldy, and is expected to contain many different concepts, each of which may on its own be reusable in other contexts, with some structures repeated within the file. The purpose of this syntactic transformation is to create ADL file(s) that act as candidate files for omic archetypes, so one large ADL file that contains several concepts would need to be divided up into smaller files, each covering a single concept without repetitions to meet the objective of facilitating reuse of archetypes for the same information in different contexts.

Alternatively, instead of creating one large ADL file from the mzML schema that needs to be divided up into smaller, non-overlapping archetypes, choosing a smaller granularity for the output files facilitates the semantic review process. Upon examination of the XML file it was thought that choosing the definition of an `xs:complexType` element might be more suitable for the granularity of archetype creation, since each `xs:complexType` element covered a different aspect of the mass spectrometry experiment, such as “chromatogram”, “sample”, or “processingMethod”. Thus, it was decided that for each rectangle in Figure 4.4, corresponding to an `xs:complexType` element definition, one ADL file should be created, acting as candidate archetype for the review process. Furthermore, many of the defined `xs:complexType` elements are reused throughout the mzML schema definition in the definition of more complex XML elements. Thus, creating one ADL file for each `xs:complexType` element defined in the schema aligns much more closely to semantic archetype design principles to design generic archetypes that can be reused in different places. While this granularity may be too fine to directly correspond to the level of granularity expected for archetypes, creating candidate archetypes at this fine level of granularity would make these ADL files easier to review in Phase 2, as the size of the resulting ADL files is small and their content clear and they can always be combined with or absorbed into other archetypes, with the aim to create reusable archetypes. The idea was that the content of the mzML file can be recomposed with the final set of reusable archetypes that are arranged in templates customised to local requirements.

To represent any hierarchical data structure in archetypes, the *openEHR* CLUS-

TER and ELEMENT classes can be used. Both CLUSTER and ELEMENT are subclasses of the abstract ITEM class and while ELEMENT is a leaf data type and thus at the bottom of the data hierarchy, a CLUSTER can contain further ITEMS, i.e. more CLUSTERS or ITEMS and thus represent an arbitrary data structure [240]. The relationship between these classes is shown in Figure 4.6.

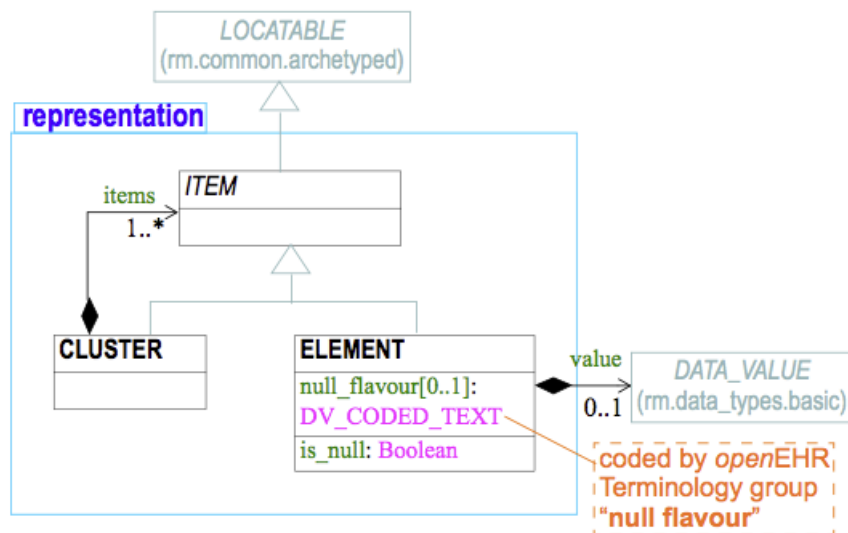


Figure 4.6: Relationship between ITEM, CLUSTER and ELEMENT, Source: [240]

The CLUSTER class is one of the most “generic” archetype classes and does not convey any meaning regarding the type of concept that it models. Thus, it was decided that for each `xs:complexType` element defined in the original data format, a CLUSTER type ADL file mirroring the content of the `xs:complexType` element is created. With each `xs:complexType` definition mapped to a CLUSTER archetype, the value of the name attribute of the `xs:complexType` definition becomes the name of the archetype in the archetype ID, with some modification: all `xs:complexType` elements have a name attribute with a trailing “Type”, so in the naming of the CLUSTER file, the trailing type should be omitted (i.e. if name=“SpectrumType”, then the name becomes “spectrum”), further, to adhere to archetype convention, names should be converted to lowercase and double words, such as “dataProcessing” should be connected with an underscore, as in “data_processing”. For example, for the `xs:complexType` with name=“SpectrumType”, the corresponding archetype ID becomes *openEHR-EHR-CLUSTER.spectrum.v1*. Further, any documentation provided by `xs:complexType` elements to describe the `xs:complexType` element is used to define the purpose in the archetype description section.

Mapping of XML data types to *openEHR* data types

The XML file references a number of data types that need to be mapped to *openEHR* data types. The possible values of the type attribute of the `xs:element` and `xs:attribute` definitions are mapped to *openEHR* data types as shown in Table 4.1.

XML data types	Corresponding <i>openEHR</i> data types
<code>xs:complexType</code>	CLUSTER
<code>xs:string</code>	DV_TEXT
<code>xs:nonNegativeInteger</code>	DV_COUNT
<code>xs:ID</code>	DV_IDENTIFIER
<code>xs:anyURI</code>	DV_URI
<code>xs:IDREF</code>	DV_EHR_URI
<code>xs:int</code>	DV_COUNT
<code>xs:dateTime</code>	DV_DATE_TIME
<code>xs:base64Binary</code>	DV_TEXT or DV_MULTIMEDIA

Table 4.1: Mapping of XML data types to *openEHR* data types

Mapping of value-type terms to *openEHR* data types

Not only the data types defined in the mzML schema can appear in the XML instance document, but the mapped CVs can also define data types. For example, the ontology term “customization” with accession number “MS:1000032” defines an element of type string via the “`xref: value-type`” attribute as shown in Listing 4.8.

```
[Term]
id: MS:1000032
name: customization
def: "Free text description of a single customization made to the instrument; for
      several modifications, use several entries." [PSI:MS]
xref: value-type:xsd\:string "The allowed value-type for this CV term."
is_a: MS:1000496 ! instrument attribute
```

Listing 4.8: An ontology term defining an element of type string

Some of the data types used in the ontology are the same as those used in the type attribute (e.g. `xsd:string` corresponds to `xs:string`) and are mapped the same way. Table 4.2 shows how any further data types only used in the psi-ms ontology were mapped to *openEHR* data types.

Value types defined in psi-ms.obo ontology	Corresponding <i>openEHR</i> data type
xref: value-type:xsd\:integer	C_DV_COUNT
xref: value-type:xsd\:positiveInteger	C_DV_QUANTITY
xref: value-type:xsd\:bool	DV_BOOLEAN
xref: value-type:xsd\:boolean	DV_BOOLEAN
xref: value-type:xsd\:decimal	C_DV_QUANTITY
xref: value-type:xsd\:date	DV_DATE
xref: value-type:xsd\:float	C_DV_QUANTITY
xref: value-type:xsd\:double	C_DV_QUANTITY
xref: binary-data-type:MS\:1000521 ! 32-bit float	DV_TEXT
xref: binary-data-type:MS\:1000523 ! 64-bit float	DV_TEXT
xref: binary-data-type:MS\:1000519 ! 32-bit integer	DV_TEXT
xref: binary-data-type:MS\:1000522 ! 64-bit integer	DV_TEXT
xref: binary-data-type:MS\:1001479 ! null-terminated ASCII string	DV_TEXT

Table 4.2: Data types specified in PSI-MS ontology

XSD internally defined types and archetype slots

An `xs:element` can refer to an internally defined `xs:complexType` element. In the case of mzML file, a reference to an `xs:complexType` element defined elsewhere in the XSD file is preceded with a “dx:”. Thus, an `xs:element` that refers to another `xs:complexType` element defined in the XSD is mapped to an archetype slot object that restricts the slot to the CLUSTER archetype defined by the referred-to `xs:complexType`. The use of CLUSTERS and the slot mechanism allows the XML instance hierarchy dictated by nesting the `xs:complexType` elements in the mzML schema (as shown in Figure 4.4) to be replicated in an archetype hierarchy by nesting the created CLUSTERS using the slot mechanism.

Mapping of `xs:complexType` element and attribute definitions to archetype objects

A `xs:complexType` elements can contain a (sequences of) sub-elements of either complex or simple type (`xs:complexType` or `xs:simpleType`, respectively) and (any number of) attribute(s). The decision was taken to treat `xs:attribute` nodes in the XML schema in the same way as `xs:elements`. An attribute can only contain one singular value type as specified by the “type” attribute in the `xs:attribute` declaration. Thus, with `xs:complexType` mapped to a CLUSTER, the order of elements and attributes defined in the `xs:complexType` element can be mapped to the ordered list of archetype objects - further CLUSTERS, ELEMENTs or slots - in the `items` attribute of type `List<ITEM>` of the parent CLUSTER archetype object, constraining *openEHR* data

types depending on the value of their “type” attribute as explained in Section 4.4.2. For example, the `xs:complexType` definition shown in Listing 4.9 becomes the ADL construct in the archetype definition section shown in Listing 4.10.

```
<xs:complexType name="DataProcessingType">...>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" name="processingMethod" type="
      dx:ProcessingMethodType">...>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required">...>
  </xs:attribute>
</xs:complexType>
```

Listing 4.9: `xs:complexType` and sub-element hierarchy

```
definition
  CLUSTER[at0000] matches { — Data Processing
    items cardinality matches {1..*; unordered} matches {
      allow_archetype CLUSTER[at0001] occurrences matches {0..*} matches {
        include
          archetype_id/value matches {/openEHR-EHR-CLUSTER.processing-method(-[a-zA-Z0
            -9_]+)*.v1/} — Processing Method
        }
      ELEMENT[at0002] matches {
        value matches {
          DV_IDENTIFIER {*} — id
        }
      }
    }
  }
}
```

Listing 4.10: ADL equivalent for `xs:complexType` definition

Here, the `xs:complexType` element was mapped to the root object of the archetype definition (`CLUSTER[at0000]`), the child `xs:element` becomes an archetype slot object allowing `CLUSTER` archetypes and the `xs:attribute` becomes an `ELEMENT` object constraining a `DV_IDENTIFIER` object, as per the mapping of XML data types to *openEHR* data types.

Mapping of `xs:complexType` naming and descriptions to archetype ontology

Each object node in an archetype needs to be accompanied by an entry in the archetype ontology, providing a name and a description of this node. For the naming of the mapped `CLUSTER` and `ELEMENT` nodes in an archetype ontology, the “name” attribute of the `xs:element` and `xs:attribute` nodes in the XSD file can be used, becoming the “text” node in the archetype ontology. Almost every `xs:attribute` node and every `xs:complexType` node in the XML schema contains an `xs:annotation` node with an `xs:documentation` node which contains a textual description of the attribute or element node in the XML schema. These can be used for the corresponding descriptions

of the object nodes in the archetype ontology. In the case of nested `xs:complexType` elements, if the nesting does not provide a description of the nested elements itself, then the description for the nested element is extracted from the description of the nested `xs:complexType` element.

Mapping of XML occurrence and use attributes to archetype object node constraints

The occurrence attributes (“minOccurs” and “maxOccurs”) of `xs:elements` in `mzML` can be mapped to object nodes constraints in the archetype as follows:

- minOccurs and maxOccurs are not specified (default = 1), this corresponds to the default “occurrences matches 1..1” in the archetype
- minOccurs=“0” and maxOccurs=“unbounded”, this translates to “occurrences matches 0..*” in the archetype, etc.

The values of the “use” attribute of `xs:attributes` can be mapped to the occurrence constraints of archetype object nodes as follows:

- “use” attribute value set to “optional” (default, if now specified) corresponds to ‘occurrences matches 0..1”
- “use” attribute value set to “required” corresponds to “occurrences matches 1..1” (default)

For determining the cardinality of container types in the archetype (i.e. for the `CLUSTER` nodes only), the individual minimum occurrences and maximum occurrences values are combined as follows. For example, if a `CLUSTER` can have an `ELEMENT` with “occurrences matches 0..*” and another `ELEMENT` with “occurrences matches 1..1”, then the two minimum occurrences give a minimum value of “1” and the two maximum occurrences give a maximum value of “*”, so that the cardinality interval is 1..* to satisfy cardinality validity rules.

Representing mapped cvParam terms in archetypes

As discussed in Section 4.2, `mzML` is to be used in conjunction with controlled vocabularies or ontologies and a mapping file controls the usage of ontology terms within an instance of the data exchange file. The mapping file for `mzML` references several CVs or ontologies: the PSI-MS (Mass Spectrometry), Brenda/BTO (Brenda Tissue Ontology), GO (Gene Ontology), PATO (Phenotypic qualities (properties) ontology) and UO (Unit Ontology). In the `mzML` schema file, the “`CVParamType`” element acts

as placeholder for the ontology terms that may occur at this location as specified by the corresponding CVMappingRule.

There are two types of terms defined in the PSI-MS CV:

- Simple text items that themselves represent possible instance values for annotation.
- Annotated data types, such as strings or integers, that in themselves act as placeholders for run-time supplied instance data.

A mapping rule referring to a text-type ontology term thus specifies (a list of alternative) text item(s) that may be used at run-time. In archetypes, the DV_CODED_TEXT data type can be used to constrain instance data to a list of alternative text items that may be used at run-time. For example, the mapping rule in Listing 4.11 maps to a list of instrument model names in the PSI-MS ontology that may be used at run-time to annotate the instrument with the particular model. How this rule may be represented within an archetype is shown in Listing 4.12.

```
<!-- instrument -->
<CvMappingRule
  id="instrumentconfiguration-must"
  cvElementPath="/mzML/instrumentConfigurationList/instrumentConfiguration/cvParam/
    @accession"
  requirementLevel="MUST"
  scopePath="/mzML/instrumentConfigurationList/instrument"
  cvTermsCombinationLogic="AND">
  <CvTerm
    termAccession="MS:1000031"
    useTerm="true"
    termName="instrument model"
    isRepeatable="false"
    allowChildren="true"
    cvIdentifierRef="MS"
  />
</CvMappingRule>
```

Listing 4.11: CV Mapping rule for instrument node

```
ELEMENT[at0001] matches { -- Instrument model
  value matches {
    DV_CODED_TEXT matches {
      defining_code matches {
        [local::
          at0002, -- Unknown instrument model
          at0003, -- AB SCIEX instrument model
          at0004, -- 4000 Q TRAP
          at0005, -- API 150EX
          ....]
        }
      }
    }
  }
```

```
}
```

Listing 4.12: ADL code representing a list of alternative text terms

In the second case, the ontology term is mapped to an object of type `ELEMENT` in the archetype constraining the *openEHR* data type that corresponds to the data type provided in the `xref` attribute of the ontology term. For example, the ontology term in Listing 4.13 becomes the corresponding archetype construct shown in Listing 4.14.

```
[Term]
id: MS:1000586
name: contact name
def: "Name of a contact person." [PSI:MS]
xref: value-type:xsd\:string "The allowed value-type for this CV term."
is_a: MS:1000585 ! contact person attribute
```

Listing 4.13: A data-type term of type string defined in the ontology

```
ELEMENT[at0001] matches { — Contact name
  value matches {
    DV.TEXT matches {}
  }
}
```

Listing 4.14: ADL code representing a data-type ontology term of type string

Each object node in an archetype needs to be accompanied by an entry in the archetype ontology, providing a term definition. For terms derived from the CVs, the “text” attribute for terms in the term definitions section can be derived directly from the term “name” and the description can be derived directly from the annotation of the term given in the original ontology (marked “def” in the ontology). Furthermore, the term identifiers used in the PSI-MS CV can be mapped to the corresponding terms used in the archetype ontology by mapping the local term codes in the archetype to the PSI-MS accession number in the `term_binding` part of the archetype. This mapping creates a link between the terms used in the archetype ontology and the external ontology. Any external terminology used in an archetype needs to be listed in the `terminologies_available` section of the archetype ontology. The example shown in Listing 4.15 illustrates how an ontology term from an external ontology may be represented in ADL.

```
definition
.....
ELEMENT[at0027] matches { — Scan window upper limit
  value matches {
    CDV.QUANTITY <
      property = <[openehr::696]>
      list = <
        [ "1" ] = <
          units = <"m/z">
        >
      >
    >
  >
}
```



```

    >
  }
}
.....
ontology
  terminologies_available = <"PSI-MS", ...>
  term_definitions = <
    ["en"] = <
      items = <
        .....
        ["at0027"] = <
          text = <"Scan window upper limit">
          description = <"The lower m/z bound of a mass spectrometer
            scan window.">
          >
        .....
      >
    >
  >
  term_binding = <
    ["PSI-MS"] = <
      items = <
        ["at0027"] = <[PSI-MS::1000500]>
        .....
      >
    >
  >

```

Listing 4.15: Term definitions and bindings for CV terms

Using this process, the created ADL files combine both the information as defined for each `xs:complexType` element in the XML schema and the mapped CV terms from the associated CV and CVMappingsRules files.

The mapping of the attributes governing the usage of CV terms in the XML instance files to *openEHR* object attributes is described below.

Mapping of CVMappingRule and CVTerm attributes to archetype object constraints

Several related attributes of the ‘CVMappingRule’ and ‘CVTerm’ govern the occurrences details of the ‘cvTerm’ in an mzML instance file, specifically ‘requirementLevel’, ‘cvTermsCombinationLogic’, and ‘isRepeatable’. These need to be mapped to corresponding object node occurrences in the archetypes. The combination of the possible values of the ‘cvTermsCombinationLogic’ and ‘requirementLevel’ attributes of the mapping rule can be represented by the first value of the occurrences interval as shown in Table 4.3.

mappingRule cvTermsCombinationLogic	attribute mappingRule attribute re- quirementLevel	openEHR object attribute
AND	MUST	occurrences matches 1..x
OR/XOR	MAY	occurrences matches 0..x

Table 4.3: CVMappingRules and object occurrences

The ‘isRepeatable’ attribute of CVTerm can be represented with the second value of the occurrences interval as shown in Table 4.4.

isRepeatable attribute in CVTerm	openEHR object attribute
true	occurrences matches x..*
false	occurrences matches x..1

Table 4.4: CVTerm attributes and object occurrences

Mapping of units to *openEHR* terminology

Some ontology terms can also have units associated with them. For example, the term “collision energy” with MS accession number “MS:1000045” describes a data value of type “float” that has the unit “electronvolt” as specified by the “relationship: has_units UO:0000266 ! electronvolt” attribute that links to the unit ontology term for electronvolt with accession number UO:0000266 as shown in Listing 4.16.

```
[Term]
id: MS:1000045
name: collision energy
def: ‘‘Energy for an ion experiencing collision with a stationary gas particle
    resulting in dissociation of the ion.’’ [PSI:MS]
xref: value-type:xsd\:float ‘‘The allowed value-type for this CV term.’’
is_a: MS:1000510 ! precursor activation attribute
relationship: has_units UO:0000266 ! electronvolt
```

Listing 4.16: A CV term with a unit

The units as specified in the ontologies can be mapped to the property and list attributes of the C_DV_QUANTITY data type.

For example, the property value of “openEHR::121” refers to the term “Energy” as defined in the *openEHR* terminology file¹¹, which has the unit “eV” for electronVolt associated with it as shown in Listing 4.17.

```
ELEMENT[at0040] matches { — Collision energy
  value matches {
    C_DV_QUANTITY <
      property = <[openehr::121]>
```

¹¹*openEHR* terminology file, <http://www.openEHR.org/releases/1.0.2/architecture/computable/terminology/terminology.xml>

```

    list = <
      ["1"] = <
        units = <"eV">
      >
    >
  >
}

```

Listing 4.17: A data type with a unit

Different units need to be mapped individually to their corresponding terms as defined in the *openEHR* terminology. Many of the units used in mzML did not exist in *openEHR* terminology, so these needed to be added manually first.

4.4.3 Automation of the syntactic transformation process

With all the necessary mappings defined, these mappings can be applied to carry out the syntactic transformation to create ADL files from the omic data formats. Since both the XML schema and ADL files are computable files, this transformation process could be automated. This automation required the development of bespoke software that could read in the source data format, apply the established mappings and output corresponding ADL files.

Briefly, software was developed to transform the XML schema into ADL files by reading in and parsing the XSD file for the data standard and outputting a set of CLUSTER ADL files based on the `xs:complexType` nodes and the mappings established above.

The software created to transform the XML schema into ADL files was written in Java (JDK 1.6.0_26). The software made use of the *openEHR* Java reference implementation¹². The Java Reference implementation of *openEHR* can be used to create an in-memory archetype as a Java object. The reference implementation also provides an ADL Serializer that can be used to flatten the archetype object into an ADL file. Using the existing serialisation functionality will automatically ensure that the output ADL files will be well-formed. Further components used are the `java.xml.parsers` library (supporting SAX and DOM parsers) to parse and process the XML schema documents (which are themselves XML documents) and the `org.geneontology.oboedit` library to extract ontology terms and child terms from the ontologies in obo format referred to in the CVmapping file.

As previously described, the archetype definition is an instance of a `C_COMPLEX_OBJECT` that constrains a certain *openEHR* reference model class. In the Java reference implementation this corresponds to the `CComplexObject` class (a subclass of

¹²*openEHR* Java reference implementation, <http://www.openEHR.org/projects/java.html>

CObject). C_COMPLEX_OBJECTs can contain C_ATTRIBUTES (either C_SINGLE_ATTRIBUTE or C_MULTIPLE_ATTRIBUTE), which constrain the attributes of the reference model class of the parent C_COMPLEX_OBJECT. The equivalent classes for these are CAttribute, CSingleAttribute and CMultipleAttribute. Thus at the root there is a CObject that can have a CAttribute child, that itself can have CObject children and so on, until the leaf nodes, which constrain *openEHR* data types. The leaf nodes correspond to the xs:elements, xs:attributes and mapped CVTerms in the source data format. To create an archetype definition with Java objects, the archetype structure is built up in a depth-first tree building fashion, following the structure of the source XSD for each xs:complexType element.

This process recreates the structure of the xs:complexType element in the definition section of the ADL file. For each object node that is being created, in parallel a term definition for the archetype ontology is created and bound to the external ontology, where one was used.

For a more detailed explanation of the Java reference implementation used in the software and the steps performed by the software in the transformation process, please refer to Appendix C.

4.4.4 Syntactic transformation of mzML

Running the transformation software over the mzML data format, the CVMMapping file and the associated ontologies resulted in the creation of 47 candidate archetypes corresponding to the 47 xs:complexType elements defined in mzML. A list of the candidate archetypes with the descriptions as provided by the documentation nodes in the XML schema file is provided in Appendix E. These files were then reviewed in Phase 2 of the omic archetype creation process.

4.4.5 Semantic review of candidate archetypes for mass spectrometry

In Phase 2, the content of each of the automatically created ADL files (the candidate archetypes) is reviewed and modified where necessary to create high-quality archetypes that fulfil the semantic requirements of archetypes, such as covering a distinct concept at the right ontological level.

As discussed in Section 4.3.2, the “Editorial checklist for clinical content review of archetypes”, provided by *openEHR*¹³ is used for guidance in this phase.

¹³Editorial checklist, <http://www.openEHR.org/wiki/display/healthmod/Editorial+checklist+for+clinical+content+review+of+archetypes>

Relevant tasks concerning the semantic review of archetype content extracted from the editorial checklist and adapted for the omic context are listed below:

- Check that the content of the archetype is consistent with its name
- Check that the scope of the archetype is for a single omic concept
- Check that the archetype’s focus is on universal use case

Furthermore, the developed archetypes need to be checked for overlap with archetypes from existing sources. In addition, one needs to think about the level in the *openEHR* archetype hierarchy that the information contained in the original mzML file should be modelled at. This includes choosing the correct reference model class for this type of information, for example, should the root of this information be modelled on the level of an ENTRY class? If so, which ENTRY class should be chosen to model this information? The decision on this has an effect on where in the hierarchy in a Composition this information is placed.

The candidate archetypes developed in Phase 1 are based on omic data formats, which have been developed by consensus by communities of domain experts for the recording of scientific experiments. However, omic data formats for recording omic experiments have not been designed according to archetypes principles. The automatic creation of CLUSTER type ADL files based on the xs:complexType elements defined in the mzML schema file achieved a syntactical transformation of the original XML schema into archetype syntax, but the content of the automatically created CLUSTERS may not cover a “single, whole, discrete, non-overlapping” concept required of high-quality archetypes [44, 38], since the information in the source schema file may not have been organised in this manner.

The syntactic transformation process in Phase 1 took a fine-grained approach by creating one CLUSTER type ADL file for each xs:complexType element defined in the XML schema instead of creating one large ADL file that replicates the complete structure of the omic data format. This means that the content covered by each candidate archetype will most likely be covering only one particular aspect, but it may be too fine-grained, so that a particular concept may be spread over several created ADL files and these may need to be combined into one ADL file or absorbed into a parent ADL file. It is the decision of the archetype editor(s) to decide what content belongs to one concept and should be recorded by one archetype or what should be recorded in separate archetypes, taking into account how they may be reused in different contexts.

In addition, since the data formats are based on XML technology, some candidate archetypes may cover aspects that refer to details that only make sense in the original XML context, such as text descriptions referring to items in terms of the original data

format. Others may model information that is modelled differently in the *openEHR* context. These ADL files need to be eliminated or modified accordingly. In addition to the above checkpoints, *openEHR* provide a table (reproduced in Appendix D) with specific points to be checked in each archetype, concerning the completion and correctness of the data elements, their constraints and the metadata supplied in the archetype. The data elements, their constraints and the metadata of the candidate archetypes created in Phase 1 are based entirely on the original source data format. It is outside the scope of this research to check for the correctness and completeness of the original source data format. Thus, assuming correctness and completeness of the source data format and transformation software, most of the items to be checked will be automatically ensured to be complete and correct in the candidate archetype files. However, candidate archetype files may be eliminated, modified, combined and absorbed during the review process, so the spot checks were carried out on the set of archetypes that resulted from the semantic review process. Only once the structures of the candidate archetypes are finalised, other checks that would not affect archetype structure, such as appropriate naming and descriptions of items etc. were carried out. Thus, the specific tasks in the review process were largely carried out on the 47 candidate archetypes created from the 47 `xs:complexType` elements defined in *mzML* and associated *CVMappingsRules* and *CVs* in the following order:

1. Resolve scope issues amongst candidate archetypes (structural modifications)
2. Resolve overlaps with existing archetypes (structural modifications)
3. Review internal archetype content (non-structural modifications)

During this process, it is important to keep the hierarchical relationships between the data to be described intact. For example, if candidate archetypes are eliminated because the information stored by the original `xs:complexType` element is not relevant in the archetype context, then other candidate archetypes that may have slots referring to these will be modified accordingly to promote consistency.

The candidate archetypes created in Phase 1 were thus reviewed, following the tasks specified in the editorial checklist where relevant and other identified tasks as listed above. The majority of modifications on the set of candidate archetypes were carried out due to scope issues - many candidate archetypes did not make sense as stand-alone archetypes because their scope was too small or generic or did not cover a domain specific concept. Examples of commonly encountered situations and how they were resolved are described in detail in the next sections. Appendix E provides a full list of the archetypes and categorises them with regard to the the actions applied to and modifications carried out on them.

To improve readability, the names of the (candidate) archetypes have been shortened by eliminating the preceding “openEHR-EHR-” and the trailing “v1.ad” within sentences, since all the (candidate) archetypes used here are using the EHR class of the *openEHR* Reference Model and are in ADL format.

Elimination of generic and auxiliary archetypes

The scope of a number of candidate archetypes was too generic, describing generic parameter groupings, or represented auxiliary elements in the original data format, such as elements used for the association of controlled vocabulary items in the instance file, rather than a domain specific concept related to the execution and analysis of an omic experiment.

For example, the `xs:complexType` element “ParamGroupType” for which the corresponding `CLUSTER.param_group` candidate archetype was created is used as “Structure allowing the use of a controlled (`cvParam`) or uncontrolled vocabulary (`userParam`), or a reference to a predefined set of these in this `mzML` file (`paramGroupRef`)”. This element and related `xs:complexType` elements allow for the run-time definition of (referenceable) lists of user-defined parameters in the source data format. These inter-related generic placeholders for unspecified data are too unspecific to be represented as archetypes and ontologically belong more on the level of the *openEHR* reference model, since they define data structures to hold as yet unspecified data rather than domain specific concepts. Similarly, the candidate archetype for the `xs:complexType` element “`cv`” (`CLUSTER.cv`) that is used in an instance `mzML` file to provide “Information about an ontology or CV source and a short ‘lookup’ tag to refer to”. In archetypes, external terminologies to which internal terms are bound are referenced in the archetype ontology section under the “`terminologies_available`” tag. Thus, these and any slots referring to them were eliminated.

When eliminating archetypes, one has to consider the consequences for recording data, i.e. can the information that was intended to be recorded through the use of these archetypes still be represented in the remaining archetypes? Upon further inspection, it was found that all the generic archetypes could be deleted without any consequential loss of information representation by replacing the slot created for the “ParamGroupType” archetype equivalent (`CLUSTER.param_group`) with a `CLUSTER` structure (occurrences matches `0..*`), representing the information of the `userParam`, which is one of the `xs:complexType` elements defined under the `xs:complexType` element for `ParamGroup`. The “`userParamType`” element in the `mzML` file has been defined to allow for “Uncontrolled user parameters (essentially allowing free text)” when there is no appropriate controlled vocabulary terms available. A free text archetype already exists: `CLUSTER.free_text.v1`. However, the `xs:complexType` element `userParamType`

and thus the resulting CLUSTER.user_param archetype are more structured than providing a free text field, which in effect allows a user to specify data types at run-time. In archetypes, this is similar to allowing any type of data to be used at run-time, for example by using the ELEMENT[atxxxx] matches {*}, which in effect does not constrain the value type of this element and can be instantiated with any *openEHR* data type at run-time.

Thus, to represent a “userParamType” element in archetypes, the following solution is proposed: a CLUSTER containing an ELEMENT constrained to a value of type DV_TEXT (to hold the “name” attribute). Then, for the type, value and unit attributes, an ELEMENT without constraint (“any”) is used, thus allowing any value and data type to be used in an instance of this archetype. This structure is shown in Listing 4.18. This CLUSTER allows arbitrary information to be recorded (essentially allowing for user parameters to be recorded in a structured way), thereby preserving all of the omic information that can be recorded in an mzML instance file in an archetype-based instance file. Ideally, all the information that needs to be recorded is predefined and archetyped to avoid a high level of flexibility that would jeopardise interoperability. However, many data formats are still quite new and may need to mature and be completed in the future. Further, it is not always possible to predefine every piece of information that may need to be recorded in a particular circumstance, requiring the use of structures that allow for information that is unspecified at run-time to be recorded.

```

definition
.....
  CLUSTER[at0065] occurrences matches {0..*} matches {
    items cardinality matches {1..1; unordered} matches {
      ELEMENT[at0066] occurrences matches {0..1} matches {
        value matches {
          DV\TEXT matches {*}
        }
      }
      ELEMENT[at0067] occurrences matches {0..1} matches {*}
    }
  }
.....

ontology
terminologies_available = <"MS", ...>
term_definitions = <
  ["en"] = <
    items = <
      ["at0065"] = <
        text = <"Additional data">
        description = <"Uncontrolled user parameters (essentially allowing free text
          ).">
      >
      ["at0066"] = <
        text = <"Description">

```



```

        description = <"Description of additional data.">
    >
    ["at0067"] = <
        text = <"Value">
        description = <"Additional data value">
    >
>
>
>
>

```

Listing 4.18: userParam as CLUSTER

Elimination of container and complementary reference type candidate archetypes

As described in Section 4.4.1, instance mzML data is organised by grouping like information together into lists under the root node of the instance document. For example, instances of “sampleType” elements describing all samples used in a mass spectrometry experiment are listed under the “SampleListType” element instance, instances of “instrumentConfigurationType” elements describing all instrument configurations used in the experiment are listed under the “instrumentConfigurationListType” element instance, and instances of “softwareType” elements describing all types of software used in the experiment are listed under the “softwareListType” element instance etc.

Complementary to these list type elements the mzML data format defines either “reference type” elements (for which candidate archetypes have been automatically created) or attributes of type `xs:IDREF` which can be used to refer to a previously defined element instance in a list type element. Thus, the information grouped under instances of the list type elements is later contextualized within the “Run” element hierarchy, using specific reference type elements or attributes that refer to the elements previously defined and listed. This situation can be compared to a recipe, where all the ingredients are listed at the start, but then are “referred to” from the recipe description itself, which describes where and how the ingredients are used, thereby putting them into context.

For example, an instance of the “softwareRefType” element is used in the “InstrumentConfigurationType” to refer to instances of “softwareType” elements listed in the instance of the “softwareListType” container element, thus putting the previously listed element instances into a meaningful context. Similarly, the attribute “sampleRef” in the “RunType” element can be used to refer to an instance of a “sampleType” element listed under the “sampleListType” instance.

It is possible to use cross-referencing within archetypes using the `DV_EHR_URI` data type to refer to instance data recorded elsewhere in the record, as shown in Listing 4.19. The automatic transformation software creates an `ELEMENT` constrained to

a value type of DV_EHR_URI for the “sampleRef” attribute defined in the “RunType” element, so the structural organisation of information in an mzML instance file could be replicated exactly with corresponding archetypes definitions.

```

definition
.....
ELEMENT[atxxxx] occurrences matches {0..*} matches { — Sample ref
    value matches {
        DV_EHR_URI matches {*}
    }
}
.....
ontology
terminologies_available = <"MS", ...>
term_definitions = <
    ["en"] = <
        items = <
            ["atxxxx"] = <
                text = <"Sample ref">
                description = <"This attribute must reference the 'id' of the appropriate
                    sample.">
            >
        >
    >
>

```

Listing 4.19: Reference type archetype used in CLUSTER.run

Similarly, the transformation software automatically created candidate archetypes for “RefType” elements that are used as wrappers for referencing to elsewhere instantiated elements, similar to the reference attributes mentioned above.

However, the automatically created candidate archetypes for these list type and corresponding reference type elements have little semantic value in themselves, the former functioning as containers for instance elements to be referred to from elsewhere in the instance file with the latter to put the listed elements into context. They are only used to allow for the non-contextual grouping of information at the start of the mzML instance file and do not represent meaningful domain specific concepts in themselves and thus should not be archetyped. This is a redundant structure, since instead of listing the elements at the start and then referring to them, they could just as well be included directly from where they are being referenced.

The hierarchy of the “Run” element in the mzML file provides a semantically meaningful organisation of data which closely aligns with how data would be organised hierarchically in an archetype context.

Thus, both the candidate archetypes for these list type elements and their complementary reference type elements can be safely eliminated by including the candidate archetypes created for the elements listed in the list type elements directly, instead of including candidate archetypes created for reference type elements. In terms of modifying candidate archetypes, this means that the inclusion constraint for slots in candidate

archetypes created for the “reference type” elements is replaced with an inclusion constraint for the candidate archetype that is referred to by the candidate archetype for the “reference type”. For example, in the CLUSTER.instrument_configuration candidate archetype, the constraint on the inclusion criteria to allow the inclusion of “CLUSTER.software.ref” is changed to allow the inclusion of the “CLUSTER.software” candidate archetype instead, thus eliminating the need for both the candidate archetype for CLUSTER.software_list archetype and the complementary CLUSTER.software.ref candidate archetypes. Similarly, ELEMENT created for reference type attributes are also changed to slots for the element they refer to, as shown in Listing 4.20, so that instance data is directly included within the correct context.

```

definition
.....
  allow_archetype CLUSTER[atxxxx] occurrences matches {0..*} matches { — Sample
    include
      archetype_id/value matches {/openEHR-EHR-CLUSTER.sample(-[a-zA-Z0-9_])*v1/}
  }
.....
ontology
  terminologies_available = <"MS", ...>
  term_definitions = <
    ["en"] = <
      items = <
        ["atxxxx"] = <
          text = <"Sample">
          description = <"Description of sample used in this run.">
        >
      >
    >
  >

```

Listing 4.20: Slot for child archetype in CLUSTER.run

This change may lead to an issue if the items listed in the container types are referred to from several different instances within the instance file. Since this information in the modified archetypes is being included directly at each node this would lead to this information being replicated in those locations. Instance files were checked for multiply referencing the same item, and some occurrences were found, specifically regarding the use of software that may be associated both with the instrument configuration and with data processing. While such an occurrence cannot always be avoided, it can be reduced in some cases by restructuring the archetypes so that the element that is referred to multiply is placed higher up in the hierarchy. In this way it can be defined to be understood to apply to all information downstream in the hierarchy, for example by using a slot for “default software” higher in the hierarchy and another slot lower in the hierarchy that allows the inclusion of software information that differs from the default to be recorded, similar to how this is dealt with in the original data format for “defaultInstrumentConfigurationRef”.

In addition, a slightly different type of list type elements are defined in the data format. These list type elements and corresponding candidate archetypes represent listings that were used within the meaningful context of the “run” hierarchy. No equivalent reference types or attributes are defined that could be used to refer to the item(s) listed in the list types. Again, these list type archetypes do not represent a domain specific omic concept, rather they fulfil the function of listing specific items. Thus, by pulling up the information contained within these list archetypes directly into the parent archetypes that contained slots for these list type archetypes, they can be safely eliminated without losing the ability to record the information modelled by them in the reduced and modified set of archetypes. For example, in the CLUSTER.spectrum archetype, the slot for CLUSTER.product_list (with occurrences restricted to 0..1) shown in Listing 4.21 can be directly replaced with a slot for CLUSTER.product (with occurrences restricted to 0..*), the archetype that was included in the CLUSTER.product_list candidate archetype through the slot mechanism, retaining the name and description of the original slot as shown in Listing 4.22.

```

definition
.....
  allow_archetype CLUSTER[atxxxx] occurrences matches {0..1} matches {
    include
      archetype_id/value matches {/openEHR-EHR-CLUSTER.product_list(-[a-zA-Z0-9-]+)*.
        v1/}
  }
.....
ontology
  terminologies_available = <"MS", ...>
  term_definitions = <
    ["en"] = <
      items = <
        ["atxxxx"] = <
          text = <"Product List">
          description = <"List and descriptions of product isolations to the spectrum
            currently being described, ordered.">
        >
      >
    >
  >

```

Listing 4.21: Slot for the CLUSTER.product_list archetype in CLUSTER.spectrum

```

definition
.....
  allow_archetype CLUSTER[atxxxx] occurrences matches {0..*} matches {
    include
      archetype_id/value matches {/openEHR-EHR-CLUSTER.product(-[a-zA-Z0-9-]+)*.v1/}
  }
.....
ontology
  terminologies_available = <"MS", ...>

```

```

term_definitions = <
  ["en"] = <
    items = <
      ["atxxx"] = <
        text = <"Product List">
        description = <"List and descriptions of product isolations to the spectrum
                      currently being described, ordered.">
      >
    >
  >
>

```

Listing 4.22: Slot for the CLUSTER.product archetype in CLUSTER.spectrum

Resolution of overlaps with existing archetypes

Since this is the first attempt (to the best of this author’s knowledge) to develop archetypes that cover the execution and results of omic experiments on patient samples, overlaps with existing archetypes were not expected on information specific to the omic domain. However, some overlap in the information modelled with the mzML data format and corresponding automatically created candidate archetypes with existing archetypes (from the *openEHR* CKM) were detected and needed to be resolved. These generally involved the concepts of biosample, professional roles and organisations and these conflicts were resolved accordingly, by reusing existing archetypes instead, where possible as described in Chapter 4. Depending on the type of overlap, different actions were taken, which are described in the following paragraphs.

For example, the CLUSTER.sample candidate archetype overlapped with the existing CLUSTER.specimen archetype from the *openEHR* CKM.

A mapping was carried out of the definition of the CLUSTER.sample candidate archetype to the definition of the existing CLUSTER.specimen archetype from the *openEHR* CKM, where possible, to expose overlaps and gaps in the existing CLUSTER.specimen archetype. This mapping and subsequent extension of the CLUSTER.specimen archetype for missing items that were represented in the CLUSTER.sample candidate archetype, but not in the existing CLUSTER.specimen archetype are detailed in Table 4.5. In this table, the left column lists the content of the CLUSTER.sample candidate archetype, the middle column lists the mappings to equivalent descriptions in the existing CLUSTER.specimen archetype and the right column lists the extensions added to the CLUSTER.specimen archetype to cover the missing items.

CLUSTER.sample candidate archetype	CLUSTER.specimen archetype from <i>openEHR</i> CKM	Extensions added to CLUSTER.specimen
Sample number (ELEMENT of type DV_TEXT)	Use definition[at0000]/items[at0002]/items[at0001] (Specimen identifier)	-
Sample state (ELEMENT of type DV_CODED_TEXT with choices of emulsion, liquid, solid, gas, solution, suspension, terms from ontology)	-	Added ELEMENT of type DV_CODED_TEXT named "Sample state" with internal text choices of emulsion, liquid, solid, etc.
Sample mass (ELEMENT of type C_DV_QUANTITY with unit grams)	Covered in definition[at0000]/items[at0027]/openEHR-EHR-CLUSTER.physical_properties.v1/definition[at0000]/items[at0020] (Weight)	-
Sample volume (ELEMENT of type C_DV_QUANTITY with unit ml)	Covered in definition[at0000]/items[at0027]/openEHR-EHR-CLUSTER.physical_properties.v1/definition[at0000]/items[at0037]/items[at0033] (Volume)	-
Sample concentration (ELEMENT of type C_DV_QUANTITY with unit gram per liter)	-	Added ELEMENT of type C_DV_QUANTITY named "Sample concentration" with appropriate units.
Sample batch (ELEMENT of type DV_TEXT)	-	Added ELEMENT of type DV_TEXT named "Sample batch"
Sample preparation (ELEMENT of type DV_CODED_TEXT or DV_TEXT)	Covered in definition[at0000]/items[at0050]/items[at0068]/openEHR-EHR-CLUSTER.specimen_preparation.v1	-
Quality of an object (ELEMENT of type DV_CODED_TEXT with choices from PATO)	Partially covered in definition[at0000]/items[at0027] (Specimen characteristic)	Added DV_CODED_TEXT with constraint definition for the PATO value set named "Physical Quality of an object" for the ELEMENT at path definition[at0000]/items[at0027] (Specimen characteristic) and added a query identifier for the PATO term in the constraint binding section

cellular_component (ELEMENT of type DV_CODED_TEXT with choices from GO)	-	Added DV_CODED_TEXT named “Cellular component” with constraint definition for the GO value set and added a query identifier for the GO term in the constraint binding section
Brenda source tissue ontology (ELEMENT of type DV_CODED_TEXT choices from BTO)	Partially covered in defini- tion[at0000]/items[at0029] (Specimen tissue type)	Added DV_CODED_TEXT with con- straint definition for the BTO value set named “Tissues, cell types and en- zyme sources” for the ELEMENT at path definition[at0000]/items[at0029] (Specimen tissue type) and added a query identifier for the BTO term in the constraint binding section
Additional data (user- Param)	-	Added CLUSTER with ELEMENT of type DV_TEXT named “Description” and ELEMENT of “Any” value type named “Value”
Sample name (ELEM- ENT of type DV_TEXT)	-	Added ELEMENT of type DV_TEXT named “Sample name”

Table 4.5: Comparison of overlapping archetypes for specimen

Thus, the CLUSTER.specimen.v1 archetype is extended with those items that are not yet covered in the existing archetype. It should be noted that most of the description of the xs:complexType “sampleType” is defined through the use of CV-Params and userParams. During the process of incorporating these terms in the candidate archetype, the very large amount of possible CV terms (>6000) from the referenced PATO, GO and Brenda Tissue Ontology made the inclusion of all the possible CV terms in DV_CODED_TEXT lists technically impossible with current means due to memory issues. However, instead of including all the possible terms inline as DV_CODED_TEXT choices (which is unsustainable as the ontological resources are changed and updated), the ADL ‘placeholder’ internal code notation, [acNNNN], is used instead. This way the complete set of values allowed as text choices in the DV_CODED_TEXT list can be defined by a single, internal ‘ac’ code. This code is then linked to query identifiers to the appropriate external ontologies in the ‘constraint_binding’ section in the archetype, which when evaluated return the possible set of values [50]. It was found that many of the items defined in the candidate CLUSTER.sample archetype were not covered in the existing CLUSTER.specimen archetype, so that the latter needed to be extended with those items. The gaps exposed in the existing CLUSTER.specimen archetype for the sample characteristics defined in the mzML-based CLUSTER.sample candidate archetype also indicate that a more detailed description of sample characteristics for the specimen entity is required in the

research context than in the clinical context, for which the existing CLUSTER.specimen archetype had been defined.

An overlap was also detected amongst the various candidate archetypes for mass spectrometer device components listed in Table 4.6 and the existing CLUSTER.device archetype.

Candidate archetype	Description of candidate archetype
openEHR-EHR-CLUSTER.component.v1.adl	-
openEHR-EHR-CLUSTER.source_component.v1.adl	This element must be used to describe a Source Component Type. [...].
openEHR-EHR-CLUSTER.detector_component.v1.adl	This element must be used to describe a Dectector Component Type. [...].
openEHR-EHR-CLUSTER.analyzer_component.v1.adl	This element must be used to describe a Analyzer Component Type. [...].

Table 4.6: Component archetypes

A comparison between these component archetypes and the existing CLUSTER.device archetype was carried out. In this case the extra information in the component archetypes that was not covered by the existing CLUSTER.device archetype after a comparison was deemed to be specific to the description of a mass spectrometry device, rather than belonging to the generic CLUSTER.device archetype.

Thus, in this case the existing CLUSTER.device was specialised to describe the specifics of a mass spectrometer spectrometer and its sub-components, resulting in four new specialisations as listed below:

- openEHR-EHR-CLUSTER.device-mass_spectrometer.v1.adl
- openEHR-EHR-CLUSTER.device-mass_spectrometer_source.v1.adl
- openEHR-EHR-CLUSTER.device-mass_spectrometer_analyzer.v1.adl
- openEHR-EHR-CLUSTER.device-mass_spectrometer_detector.v1.adl

Then, to describe a mass spectrometer and its components with archetypes, the CLUSTER.device-mass_spectrometer archetype contains slots for the three components of a mass spectrometer.

A different strategy again was applied to the overlap between the CLUSTER.software candidate archetype and the CLUSTER.device_details archetype. A reference to CLUSTER.software was also found in the CLUSTER.data_processing candidate archetype. Thus, it was decided that the concept “software” should be described with its own archetype. This decision is justified also through the following logic: Software can be associated with a specific medical/research device, i.e. it is necessary

to configure and run the device, or it can be meaningful as a stand-alone archetype, without association to a device, specifically where the software is used to process information and the device running the software is irrelevant. Therefore, the existing CLUSTER.device_details archetype was modified by extracting the concept “software” into its own archetype containing generic information regarding software. Ultimately, the automatically created CLUSTER.software archetype was modified to represent a specialisation of the new generic CLUSTER.software archetype (CLUSTER.software-mass_spectrometry), since it contained mass spectrometry specific information.

Furthermore, the “contact” information in the CLUSTER.run candidate archetype could be covered with existing archetypes, such as CLUSTER.individual_professional.

Checking for correct data types

Some candidate archetypes were found to describe information for which native archetype data types could be used instead. For example, CLUSTER.source_file describes a mass spectrometry source file, and some of the information explicitly described in the archetype can be described combined with the *openEHR* data type DV_MULTIMEDIA instead. Table 4.7 shows which information of the CLUSTER.source_file candidate archetype can be mapped to the DV_MULTIMEDIA data type. Not all the information could be mapped, only that generic to describing multimedia data, while the rest is specific to mass spectrometry. Thus, the information that could be mapped was replaced with the DV_MULTIMEDIA data type.

CLUSTER.source_file	DV_MULTIMEDIA
mass spectrometer file format	-
data file checksum type (MD5, SHA-1)	integrity_check_algorithm: CODE_PHRASE
data file checksum	integrity_check
native spectrum identifier format	-
name	-
id	-
sourcefile binary	data: Array <Octet>
location	uri: DV_URI

Table 4.7: Comparison of CLUSTER.source_file vs. DV_MULTIMEDIA

Similarly, the CLUSTER.binary_data_array candidate archetype, describes the information pertained in a mass spectrum or mass chromatogram, which are in effect histograms. In mzML, the x/y data points of the original histograms are encoded into two Base64 binary strings, one for the array of x-data points and one for the corresponding array of y-data points. Using this information, spectra can be recreated from these data points. During the automatic transformation, DV_TEXT was used to

hold the Base64 encoded binary string. The CLUSTER.binary_data_array candidate archetype models all the information needed to reconvert the binary data strings back into arrays of individual data points. This way of encoding the histogram data of the resulting mass spectra and chromatograms has been developed specifically for use within the mzML data format.

Some of the data the CLUSTER.binary_data_array candidate archetype models can be modelled with the *openEHR* DV_MULTIMEDIA data type, as shown in a mapping of the information provided in the CLUSTER.binary_data_array candidate archetype and the *openEHR* data type DV_MULTIMEDIA in Table 4.8. This data could be replaced with an ELEMENT with its value constrained to the DV_MULTIMEDIA data type.

CLUSTER.binary_data_array	DV_MULTIMEDIA
Binary Data Array Type (e.g. m/z array, intensity array, ...)	——
Unit (e.g. m/z, number of counts, ...)	——
Binary data type (e.g. 32-bit integer, 64-bit float, ...)	media.type: CODE_PHRASE (new entry required in terminology.xml file)
encoded length	size: Integer
binary	data: Array <Octet>
Binary compression type	compression_algorithm: CODE_PHRASE

Table 4.8: Comparison of CLUSTER.binary_data_array vs. DV_MULTIMEDIA

Alternatively, all of the data in the CLUSTER.binary_data_array archetype could be provided in form of the original XML type and the slots for the CLUSTER.binary_data_array archetype could be replaced with an ELEMENT with its value constrained to the DV_PARSABLE data type, which is a *openEHR* data type used to encapsulate parsable string type data, such as XML.

It was decided that the latter presented the most reasonable solution, since the information defined in the original xs:complexType element and the corresponding candidate archetype is purely used to be able to reconvert the binary data values into the original histograms. In fact, the information recorded with this element is most likely recorded directly from the device without human intervention, and will also not be assessed by humans, but directly by another device, which means that there is also no need to provide a human readable display on the screen of some of the raw data recorded such as the individual m/z values recorded in the mass spectrometry experiment. Thus, the CLUSTER.binary_data_array was eliminated and slots that included this candidate archetype were changed to ELEMENTs constrained to a value of type DV_PARSABLE.

Absorption of content into parent archetypes

Some candidate archetypes were considered too small in scope to justify them as stand-alone archetypes, and their content could be considered as belonging to the concept of the parent archetype that contains a slot for it. For example, the content of the `CLUSTER.processing_method` candidate archetype is part of the concept covered by `CLUSTER.data_processing`. To recognise if the content of the child archetype could be safely absorbed into the parent archetype, all archetypes that referred to the child archetype were identified. If there was only one archetype that contained a slot for the child archetype, as in the case for the slot for `CLUSTER.processing_method` in the `CLUSTER.data_processing` archetype, then the content of the child archetype could safely be absorbed into the parent archetype, since it is not anticipated to be reused on its own in future omic archetypes, but only ever as part of the parent archetype. In these cases, the slot for these archetypes in the parent archetype was replaced with a `CLUSTER` directly representing the content of the archetype in that slot.

Furthermore, it is argued that the scope of the `CLUSTER.source_file` candidate archetype is too small to warrant a stand-alone archetype. Thus, the information modelled in the `CLUSTER.source_file` was absorbed into the parent archetype(s) that contain a slot to `CLUSTER.source_file`.

These archetypes are:

- `openEHR-EHR-CLUSTER.scan_settings.v1.adl`
- `openEHR-EHR-CLUSTER.spectrum.v1.adl`
- `openEHR-EHR-CLUSTER.scan.v1.adl`
- `openEHR-EHR-CLUSTER.precursor.v1.adl`
- `openEHR-EHR-CLUSTER.run.v1.adl`

By absorbing the content of the child archetypes into the parent archetypes these archetypes whose scope is too small could be eliminated. Some of the container type archetypes that contained some more information than simply a slot of the listed child archetype also fell into this category. The decision of absorbing content of a candidate archetype into a parent archetype also affected archetypes for which no reusability outside of a mass spectrometry experiment could be foreseen. This criteria led to the absorption of a further set of candidate archetypes, for example of the `CLUSTER.scan` archetype was absorbed into the parent `CLUSTER` candidate archetype.

Choosing appropriate root archetype class

For the automatic transformation process it was decided to create a CLUSTER type candidate archetype for each xs:complexType element defined in the source data format to facilitate the review process in Phase 2, including the root archetype class. However, the CLUSTER archetype may not be the most appropriate archetype class at which the information of a mass spectrometry experiment should be modelled at. Possible choices of “archetypeable” *openEHR* reference model classes, in hierarchically descending order are:

- COMPOSITON
- SECTION
- ENTRY (with subtypes OBSERVATION, EVALUATION, INSTRUCTION, ACTION, ADMIN_ENTRY)
- STRUCTURE (with subtypes ITEM_LIST, ITEM_SINGLE, ITEM_TABLE and ITEM_TREE)
- CLUSTER
- ELEMENT

These classes and their hierarchical relationships have previously been discussed in Chapter 2. As a reminder, a COMPOSITION corresponds roughly to a clinical document, a SECTION to a document heading, and ENTRYs to single clinical statements. STRUCTUREs can be used to explicitly model logical data structures, such as lists, single items, tables and trees. CLUSTERs can be used to hierarchically group ITEMs. ELEMENTs are leaf variants of ITEM types, and instances of ELEMENTs contain the actual data value instances. The first decision that needs to be taken is on which level of the reference model hierarchy the information regarding a mass spectrometry experiment should be recorded. In other words, what is the appropriate archetype level equivalent for the information modelled in the mzML data format. It was decided that the most appropriate level for this information would be on the level of an ENTRY type archetype, because “an ENTRY is the root of a logical item of “hard” clinical information created in the “clinical statement” context, within a clinical session” [42], in the clinical context. The second decision that needs to be taken is which ENTRY type sub-class should be used for this information. Applying the decision algorithm provided by *openEHR*¹⁴ in the omic context, it was found that the information de-

¹⁴Decision algorithm, <http://www.openEHR.org/wiki/download/attachments/786529/Decision+algorithm.gif?version=3&modificationDate=1195620505000>

scribed by mzML, which provides a way of recording a mass spectrometry experiment and result, corresponds most closely to the level of an OBSERVATION.

Thus, the information modelled in the CLUSTER.run archetype was reorganised into the OBSERVATION archetype, which required a restructuring of the content and organisation of data in the archetype definition to adhere to the class structure of the OBSERVATION class. These changes are explained in the following paragraphs.

The OBSERVATION archetype class organises information under data, state and protocol attributes. In particular, the data attribute contains the actual data being recorded, as a History of Events, so it is here where the mass spectrum and/or mass chromatogram data should be recorded. “Any particular information about the state of the subject of the Entry necessary to correctly interpret the data” can be recorded either in the state attribute of the OBSERVATION, or alternatively “within the state attribute of each Event in the Observation data attribute” [42]. In the case of a mass spectrometry experiment, each mass spectrum/chromatogram has different state information associated with it, therefore the latter case would ideally be used and state is recorded with every data item. This includes precursor and product isolation settings and the settings for each individual scan. Finally, the protocol section in the OBSERVATION can be used to record information about the device used, specimen details and contact information. Thus effectively, CLUSTER.run archetype was reorganised to become the OBSERVATION.mass_spectrometry archetype.

Editorial review of content of remaining archetypes

The remaining archetypes were reviewed according to the “Editorial checklist for clinical content review of archetypes”¹⁵. For example, in several instances, the text in the descriptions of the archetype or of specific nodes was changed, for example where the text specifically referred to XML elements or XML technology. The purpose of the CLUSTER.device-mass_spectrometer_source archetype was changed from the text provided for the SourceComponentType element in the mzML file (“This element must be used to describe a Source Component Type. This is a PRIDE3-specific modification of the core MzML schema that does not have any impact on the base schema validation”) to “To record the details of a mass spectrometer source component”. In addition, some of the meta-data was completed where appropriate, e.g. a purpose, use, and keywords were added. These items needed to be completed manually, since this information was not provided in the original data formats. Similarly, units needed to be matched to adhere to the unit descriptions in the *openEHR* terminology support

¹⁵Editorial checklist, <http://www.openEHR.org/wiki/display/healthmod/Editorial+checklist+for+clinical+content+review+of+archetypes>

file¹⁶. In addition, the units specific for mass spectrometry data needed to be included in the *openEHR* terminology file.

Execution of review process and archetype modifications

While the above description of the review process may give the impression that it was carried out in a linear manner, in practice the process is much more iterative, with early decisions taken of how to reorganise the information often leading to dead-ends, until a satisfactory solution was found that would satisfy semantic archetype requirements while covering all the information as described in the consensus-based original data formats for a mass spectrometry experiment.

During this review process, the content of archetypes often needed to be rearranged involving several archetypes simultaneously, rather than only reorganised internally within a single archetype at a time. Current archetype editors only support working on a single archetype in isolation. This means that when content from one archetype needs to be moved into another archetype, this process involves the copying of several parts of the source ADL file into the target ADL file: including the selected archetype node hierarchy in the archetype's definition section, the corresponding term definitions in the archetype's ontology section, and possibly further term bindings in the archetype's ontology section, which are all related through the archetype node identifier. All these fragments need to be copied into the appropriate sections in the target archetype while maintaining the uniqueness of node identifiers in the target archetype. Since archetype node identifiers are only unique within each archetype and generally follow an incremental numbering scheme (the first node generally starts with 'at0000' and subsequent nodes increase incrementally, i.e. 'at0001', 'at0002', 'at0003', etc. although the order is irrelevant and not strictly followed), two different archetypes would have greatly overlapping archetype node identifiers.

Thus, before any fragments are copied from the source into the target archetype, the uniqueness of node identifiers over the set of archetypes across which the contents are to be rearranged needs to be ensured first. This can be done, for example, by renumbering all corresponding occurrences of nodes across the several sections, starting each archetype at a higher number than the highest one in the set of archetypes to be arranged (while maintaining node identifier integrity within the archetype), to achieve global uniqueness amongst the node identifiers of the set of archetypes to be rearranged. To support the renumbering of the archetypes, a perl script was developed that carried out the renumbering of corresponding nodes in the definition and ontology sections starting from any chosen number and incrementing for subsequent

¹⁶ *openEHR* Terminology Support file, available at <http://www.openehr.org/releases/1.0.2/architecture/computable/terminology/terminology.xml>

nodes in the ontology section to ensure global uniqueness of archetype node identifiers across the set of archetypes whose contents needed to be rearranged, before copying and pasting ADL fragments between archetypes. This script could then also be applied after fragments of the source archetype had been copied into the target archetype to achieve incremental node identifiers as they appear in the ontology section (although this is not a requirement). Once uniqueness of nodes across affected archetypes has been carried out, the content of the source archetype can be moved into the target archetype without upsetting archetype validity. However, since the current archetype tooling does not support a drag and drop functionality of nodes and corresponding term definition and term binding (and possibly further) sections between two or more archetypes, this process of moving part of one archetype hierarchy into another had to be carried out using the copy and paste functionality of a text editor directly on the ADL files themselves. This manual process of combining content from two archetypes into one is error prone and tedious and great care needs to be taken to not upset the integrity and validity of the resulting archetypes.

It would have been useful if there existed an archetype editor that supported the modification of two (or more archetypes) at the same time and allowed for dragging and dropping fragments of one archetype into another. Such drag and drop functionality of archetype nodes in the definition section of an archetype in ADL format should also support the dragging and dropping of the corresponding fragments in the other sections (such as the ontology section), while maintaining the integrity of node identifiers in the different sections of the ADL files.

Further, to minimise having to carry out manual modifications of ADL files in a text editor to avoid aforementioned difficulties, changes to the ADL files were not carried out manually in a text editor, where possible. Instead, the transformation software was modified and re-run to create the modified archetypes, especially when the same modification needed to be carried out in several archetypes. This meant, for example, that the uniqueness of the node IDs in the candidate archetypes could be assured automatically, thus avoiding having to renumber them manually whenever modifications have been carried out that compromised uniqueness of node IDs. Thus, the semantic refinement process was somewhat of a learning exercise that has resulted in improved iterations of the transformation software. As a result, it would now take much less time to re-generate the same archetypes if starting anew, since many of the semantic modifications that needed to be carried out are already incorporated into and carried out by the transformation software.

In summary, a semi-automatic approach was developed that was used successfully to create archetypes to describe a mass spectrometry experiment from the an existing data format for mass spectrometry, mzML.

Resulting set of omic archetypes for mass spectrometry

The review process reduced the initial number of 47 candidate archetypes to a revised set of 10 semantically sound omic archetypes, as listed in Table 4.9, that can be combined in a template with a number of existing archetypes, as listed in Table 4.10, to describe a mass spectrometry experiment and the results. The archetypes in ADL format are provided on the CD supplied with this thesis.

A list of the candidate archetypes and the modifications carried out on them is provided in Appendix E.

Final archetype	Description
CLUSTER.device.details-mass_spectrometer.v1	To record additional details for the running of a mass spectrometer.
CLUSTER.device-mass_spectrometer_analyzer.v1	Use to record the details about a mass analyzer (or mass filter) component of a mass spectrometer.
CLUSTER.device-mass_spectrometer_detector.v1	To record the details of a detector component of a mass spectrometer.
CLUSTER.device-mass_spectrometer_source.v1	To record the details of a mass spectrometer source component.
CLUSTER.device-mass_spectrometer.v1	Description of a particular hardware configuration of a mass spectrometer. Each configuration must have one (and only one) of the three different components used for an analysis. For hybrid instruments, such as an LTQ-FT, there must be one configuration for each permutation of the components that is used in the document. For software configuration, use CLUSTER.device_details.v1.
CLUSTER.mass_chromatogram.v1	To describe a single chromatogram.
CLUSTER.mass_spectrum.v1	The structure that captures the generation of a peak list (including the underlying acquisitions). Also describes some of the parameters for the mass spectrometer for a given acquisition (or list of acquisitions).
(CLUSTER.software.v1)	(Software information.)
CLUSTER.software-mass_spectrometry.v1 (specialization of new openEHR-EHR-CLUSTER.software.v1.adl archetype)	To describe mass spectrometry software.
OBSERVATION.mass_spectrometry.v1	To describe a run in a mass spectrometry analysis. A run in a mass spectrometry analysis record should correspond to a single, consecutive and coherent set of scans on an instrument. Captures the use of a mass spectrometer, the data generated, and the initial processing of that data (to the level of the peak list).

Table 4.9: List of final mass spectrometry archetypes

Existing archetype	Description
CLUSTER.address.v1	To record details of one or more personal addresses for an individual.
CLUSTER.device_details.v1	To record additional details related to specifications, maintenance, servicing, cleaning and calibration for any device.
CLUSTER.device.v1	To record the details of a physical object, product or device used in the broadest description of the provision of healthcare but without direct therapeutic or metabolic means.
CLUSTER.individual_professional.v1	To record professional demographics including name(s), Address(es) or telecoms contacts for a person not uniquely identified within the electronic health record.
CLUSTER.organisation.v1	To record details of an organisation, for example, a third party such as a care agency.
CLUSTER.person_name.v1	To record the personal name of a patient, relative, healthcare provider or other third party.
CLUSTER.professional_role.v1	To record the role of a professional in care of an individual.
CLUSTER.telecom_details.v1	To record telecommunications details for an individual or organisation.
CLUSTER.specimen.v3 (modified)	To record details of a laboratory specimen. Will often be used in different contexts e.g within an Instruction archetype to describe the specimen that has to be taken, or describing the specimen which accompanies the laboratory request. It may occur within an Action archetype e.g. describing specimens taken as part of a surgical procedure. It will finally be used within a laboratory result archetype to describe the specimen being reported.
COMPOSITION.report.v1	Generic container archetype to carry information that needs to be shared with others.

Table 4.10: Existing archetypes needed for full mass spectrometry template

Many other omic data formats have been developed/are currently being developed as described in Chapter 2, which may be used as the basis for the creation of further omic archetypes. The next section describes the results of applying this 2-phase process to other omic data formats.

4.5 Development of omic archetypes for other omic data formats

4.5.1 Introduction

The previous section detailed the development of generic omic archetypes modelling the execution and results of mass spectrometry experiments, based on the mzML data

format, using a novel two-phase semi-automatic approach.

As discussed in Chapter 2, mzML is but one of many omic data formats that the omic community developed to enable the consistent recording of the execution and results of an omic experiment in compliance with the minimum information reporting standards in an interoperable way so that they can be shared safely and interpreted correctly. While these data formats can be used to record information that complies with MI reporting standards or guidelines, they generally allow for more or less information than is required by the MI reporting standards to be recorded. Thus, the 2-phase omic archetype development approach demonstrated on the example of mzML to develop archetypes for mass spectrometry may be applied to other omic data formats. As discussed further in Chapter 2, there are increasingly efforts to consolidate and coordinate amongst the different standard development initiatives. On the level of data formats, the FuGE initiative has been set up to provide a generic data model to facilitate the development of further data standards in functional genomics. FuGE can be used on its own to record a simple experiment, but in general, extensions to the FuGE object model are/have been developed that provide a more specific context. Many of the newer (versions of) omic data formats are now based on the FuGE object model, e.g. the proteomic data formats GelML and spML and others. Due to its importance and its influence on future omic data formats, the two-phase omic archetype development approach used for the development of generic mass spectrometry archetypes was next applied to FuGE and one of its extensions, GelML. By adapting the process to the FuGE data format and GelML as a representative of one of its extensions, it can easily be applied to other FuGE based formats. This section describes the process and results of applying the above developed semi-automatic two-phase approach to FuGE and one of its extensions, GelML, thus providing an assessment of the genericity of this approach.

4.5.2 The FuGE data exchange format

The structure of the FuGE data format is discussed briefly in this section. FuGE is available as a UML model and every object of the FuGE model descends from one or both of two hierarchical base classes: “Describable” and its subclass “Identifiable”, as shown in Figure 4.7

The structure of FuGE has been explained briefly in Chapter 2, and is reproduced here for convenience: FuGE is comprised of 10 packages divided into two categories [129, 138] as shown in Figure 4.8. The “Common” package and its sub-packages are used for general data standards functionality [129, 138]. The “Bio” package and its sub-packages contain modules for describing experimental procedures, such as experimental design and sample annotation [129, 138]. FuGE can be used to encode biological workflows,

such as experimental procedures or methods, SOPs, instrumental or software analysis procedures with the “Protocol” package [138]. The “Collection” package defines the root structure of the XML format and is used in an instance document to group all objects within a FuGE instance [223].

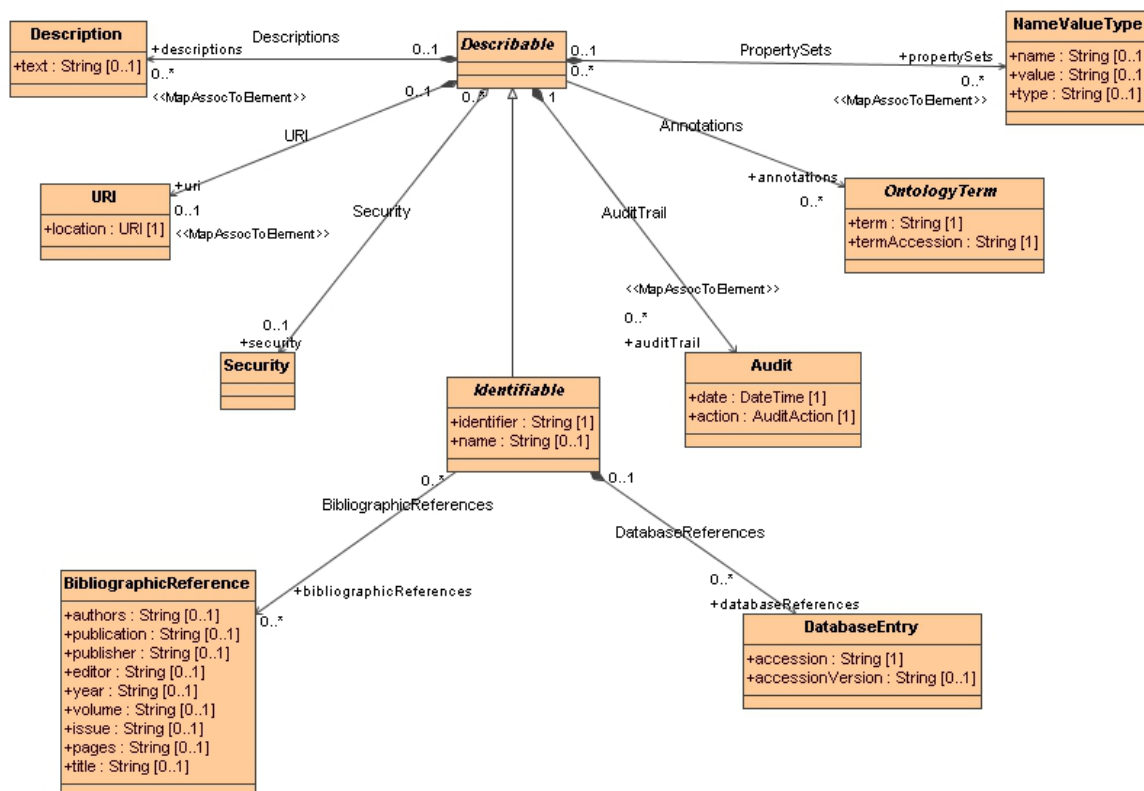


Figure 4.7: The hierarchy of base classes in FuGE from the FuGE specification, Source: [223]

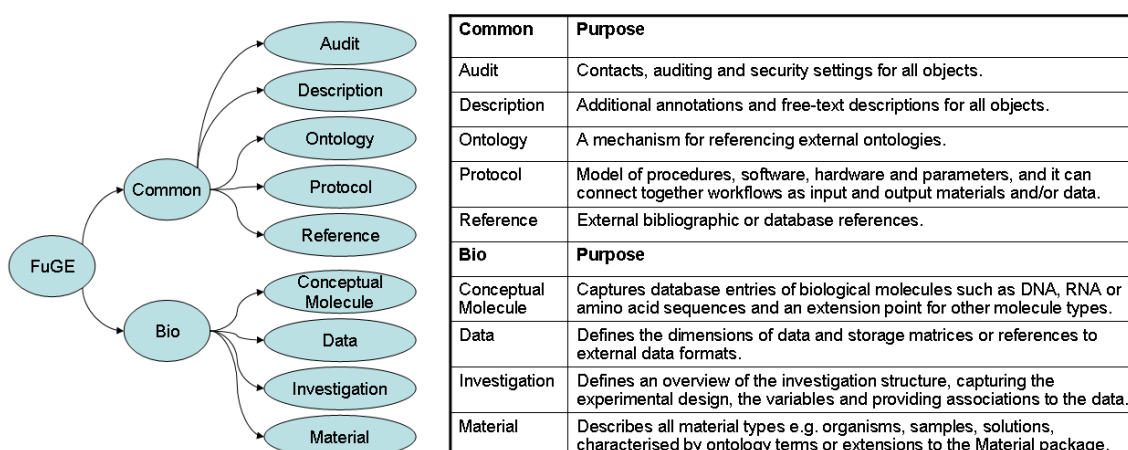


Figure 4.8: The hierarchy of base classes in FuGE from the FuGE specification, In addition to Common and Bio, a third package exists as a child of FuGE, Collection, which contains only classes and no child packages, Source: <http://fuge.sourceforge.net/>

As shown in Figure 4.9, the FuGE object model is organised with the FuGE class at the top of a hierarchy that contains all classes in the FuGE object model through composite associations.

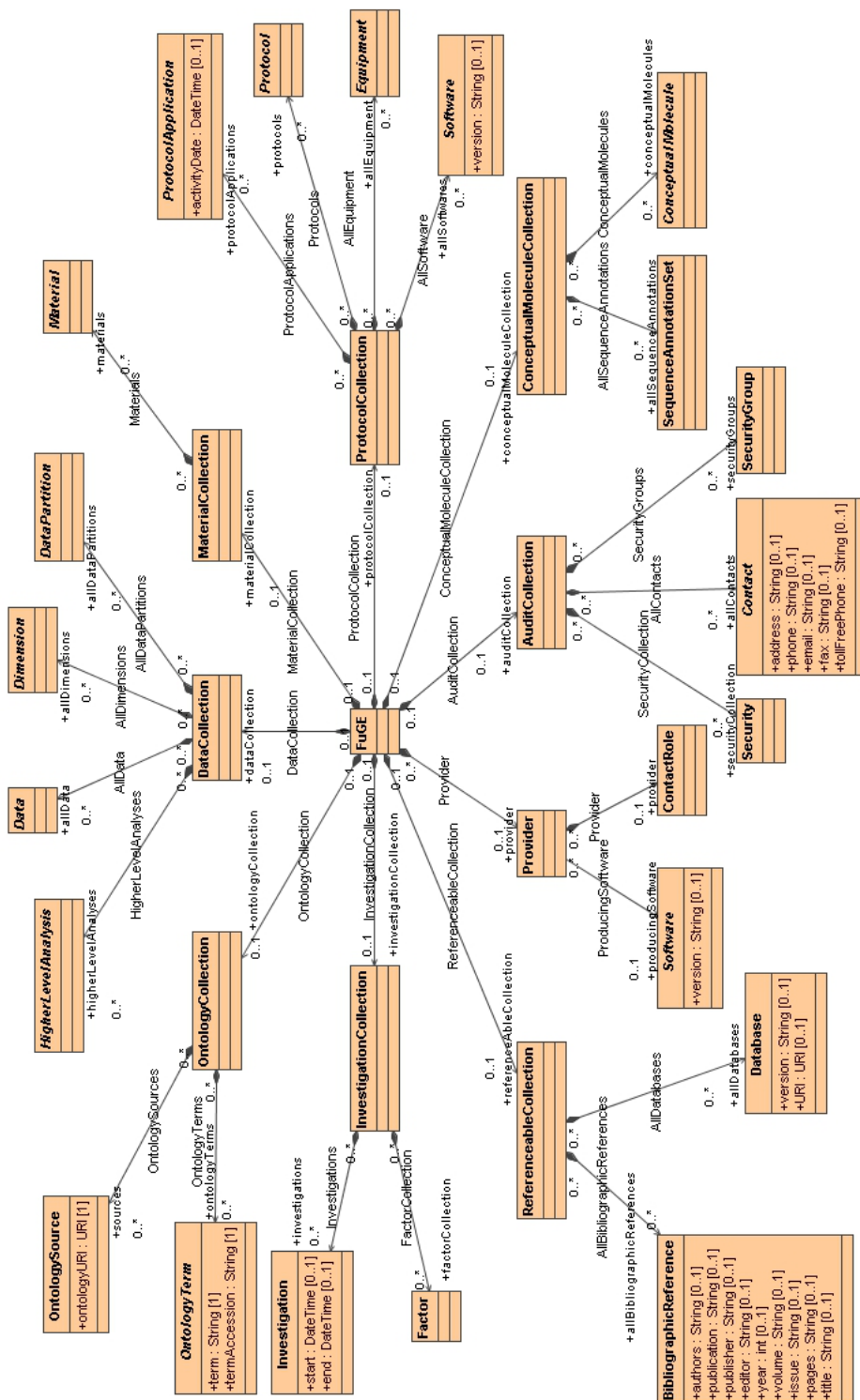


Figure 4.9: The FuGE class defines the root of a document and has, directly or indirectly, composite associations to all classes in the object model, Source: [223]

4.5.3 Syntactic transformation of FuGE

The FuGE UML model has been mapped to an XML schema¹⁷, which was used as the source format for the two-phase omic archetype development approach. Each FuGE UML class is represented by an `xsd:complexType` element definition in the FuGE XSD file. As for mzML, in the first phase, a syntactic transformation is carried out that uses the FuGE XML schema as source format for the creation of a number of candidate archetypes. Both mzML and FuGE are expressed as XML schemas, so the same mappings defined for mzML were applied and the software developed for the transformation of mzML into CLUSTER type candidate archetypes was reused.

However, due to a number of differences in how the FuGE-based schema defines `xs:complexType` elements in comparison to the mzML schema, the automatic transformation software needed to be adapted slightly. The main modifications needed for the software to work for the FuGE-based data formats are described below.

Adaptation of automatic transformation software to FuGE and FuGE-based omic data formats:

1) Adapting to different namespace: The FuGE and FuGE-based schema files (e.g. spML, GelML) use the “xsd:” prefix rather than the “xs:” prefix used in mzML, so all references in the software to the namespace prefix were changed to “xsd:”.

2) Adapting to different placeholders for external ontology terms: The placeholders for inclusion of external ontology terms are specified differently in FuGE. Instead of the `CVParamType` element definition used in mzML, in FuGE, the placeholder for an ontology term is defined by an unnamed `xsd:complexType` element that contains an `xsd:attribute` with `name= “OntologyTerm_ref”` of `type= “xsd:string”`, expecting a value of type string from an ontology in the instance file. As previously discussed in the section on CVParams in mzML, if the ontology is used for providing string values for annotation only rather than for the definition of new data types, the `DV_CODED_TEXT` data type can be used with the annotations coded as local data values or with the `[acNNNN]` construct. Therefore, the transformation software was adjusted accordingly. Upon encountering an `xsd:complexType` element that contained an `xsd:attribute` with `name= “OntologyTerm_ref”`, the transformation software created an `ELEMENT` constraining a `DV_CODED_TEXT` with Codephrases representing the allowed ontology terms (according to accompanying `CVMappingsRules`) in the candidate archetype.

Although placeholders for ontology terms were defined in the FuGE schema, no CV

¹⁷FuGE UML model, <http://fuge.sourceforge.net/>

mapping file is associated with the FuGE schema, thus no terms could be included automatically in the created archetypes. While there is no CV mapping file associated with FuGE, FuGE does depend on the use of external ontologies or controlled vocabularies. The FuGE creators provide recommendations of using the OBI ontology in conjunction with FuGE to describe experimental procedures, investigation designs and descriptors, security settings, biological sequence types, units and simple data types and some characteristics of biological samples and materials. Other species-dependent ontologies may be used to describe source samples and materials in conjunction with OBI [223].

3) Adapting to structural differences: There are also some structural differences in how FuGE defines sub-elements and attributes within an `xsd:complexType` element. In `mzML`, `complexType` sub-elements of type `xs:element` are listed under an `xs:sequence` node and `xs:attributes` are listed directly under the parent `xs:complexType` node. Further, the type of a complex sub-element is specified through the “type” attribute in the `xs:element` tag (such as `<xs:element name=“fileContent” type=“dx:ParamGroupType” >`), referring through the “dx:” namespace to an elsewhere defined `xs:complexType` element internally. In contrast, in FuGE, an “`xsd:sequence`” element is used to list sub-elements and attributes.

The sub-elements are defined in two different manners. In the first case it is defined as an element that refers to an “`xsd:complexType`” element defined elsewhere through the “ref” attribute of the sub-element to represent composite (i.e. containment) associations (e.g. `<xsd:element ref=“fuge:InvestigationComponent” ...>`). In the second case the sub-element specifies the type through a child `xsd:complexType` node, which has children of type “`xsd:attribute`” for simple types such as “`xsd:string`” or an “`xsd:element`” with a “ref” attribute referencing an elsewhere defined element in the “fuge:” namespace, as before. The transformation software was modified to deal with this different structure accordingly and so that it creates an archetype slot by default for any element that contained a reference to another element (through the “ref” attribute, e.g. as in `<xsd:element ref=“fuge:InvestigationComponent” ... >`, which describes associations with the UML object named in the ref attribute).

Listing 4.23 shows an `xsd:complexType` element definition in FuGE and Listing 4.24 shows the corresponding ADL structure as output from the modified software.

```
<xsd:complexType name=“FuGE.Common.DescribableType” abstract=“true”>
  <xsd:annotation>
    <xsd:documentation> documentation text </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name=“_auditTrail” minOccurs=“0”>
      <xsd:annotation>
```

```

        <xsd:documentation> A list of Audit instances that track changes to the
            instance of Describable.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="fuge:Audit" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
.....
</xsd:element>

```

Listing 4.23: An xsd:complexType element declaration in FuGE

```

.....
definition
  CLUSTER[at0000] matches {
    items cardinality matches {0..*; unordered} matches {
      allow_archetype CLUSTER[at0001] occurrences matches {0..*} matches {
        include
          archetype_id/value matches {/openEHR-EHR-CLUSTER.audit(-[a-zA-Z0-9-]+).v1/}
        }
      }
    }
  }
.....
ontology
  terminologies_available = <...>
  term_definitions = <
    [ "en" ] = <
      items = <
        [ "at0000" ] = <
          text = <"Describable type">
          description = <"documentation text ">
        >
        [ "at0001" ] = <
          text = <"Audit Trail">
          description = <"A list of Audit instances that track changes to
            the instance of Describable. ">
        >
      >
    >
  >
.....
>

```

Listing 4.24: ADL output for FuGE.Common.DescribableType

In addition, non-composite associations are modelled in FuGE to be included on the instance level through attributes with names ending in “_ref”, for example `<xsd:attribute name="Equipment_ref" type="xsd:string" use="required">`.

In mzML, it was possible to create a true, containment hierarchy by eliminating list types and replacing instance level referencing (the mzML equivalent of non-composite associations) with containment associations (through using the slot mechanism or including the referenced information directly at the referenced location to include the previously referenced item directly in a containment hierarchy), without losing meaning. This was possible, because the original containment hierarchy (the list type elements) does not add context by itself, but because information was put in context

through the instance level references and information could be organised in a suitable information hierarchy that avoided the instantiation of repeat information.

The modelling of information in the FuGE data format is much more complex. It provides container types with containment hierarchies whose content is referenced by pointer from within a contextual hierarchy. In addition, there is a much higher level of non-composite associations, and one set of information may be associated with several other sets of information. In other words, a set of information instantiated once (as part of the containment hierarchy) may have multiple non-composite associations in addition to the composite association. Container types (“Collections”), which provide the containment hierarchy for many sets of information are eliminated, so when these sets have several non-composite associations, one needs to decide which non-composite association can be changed into a containment hierarchy, or, in other words, under which archetype that refers to this set should this information be included (either directly, if the scope is too small or through the slot mechanism, if the information is represented with its own archetype). This decision largely depends on the scope of the particular archetypes that references the particular information and therefore is left to the review phase. For composite (i.e. containment) associations, slots are created as in mzML, whereas for non-composite associations, an ELEMENT restricted to value DV_EHR_URI is created instead of a DV_TEXT element to model this relationship.

Furthermore, FuGE encodes the complete package structure in the names of the classes that are represented as “xsd:complexType” elements, as a concatenated, dot-separated string, such as “FuGE.Common.DescribableType”. For the naming of the candidate archetypes and internal archetype nodes, the names were shortened by deleting the preceding concatenated package structure of FuGE nodes and eliminating the trailing “Type”. For example, “FuGE.Common.DescribableType” simply becomes “Describable”. In fact, in the FuGE schema, all elements defined in the schema are listed at the end of the XSD file, and the name attribute represented the shortened name, so that it could be used directly for the naming of the candidate archetypes (after inserting the underscore (‘_’) to divide words and decapitalisation).

For example, the element listed at the bottom of the FuGE schema file as shown in Listing 4.25 has the name attribute set to “DatabaseReference”, which is used to name the archetype created for an element of this name and of the specified type to CLUSTER.database_reference.

```
<xsd:element name="DatabaseReference" type="fuge:FuGE.Common.References.  
DatabaseReferenceType" abstract="false" substitutionGroup="fuge:Describable"/>
```

Listing 4.25: Element list extract

The modified transformation software created 77 CLUSTER type ADL files representing the 77 “xsd:complexType” elements defined in FuGE, which correspond to

the 77 FuGE UML classes. A list of these 77 candidate archetypes is provided in Appendix F.

4.5.4 Semantic review of candidate archetypes for functional genomics

As for mzML, the automatically created candidate archetypes for FuGE need to be reviewed as part of the second phase of the omic archetype development. During the review of the 77 candidate archetypes that have been created automatically from the FuGE data format representing FuGE UML classes, it was found that most of them are not suitable as archetypes, generally because their scope was too generic or too low-level. Thus, all the candidate archetypes that are too generic or low-level are identified and eliminated with the remaining candidate archetypes to be reviewed in depth.

Whole groups of candidate archetypes are eliminated due to similar characteristics. A short summary of the characteristics of the archetypes that are eliminated and their reasons is provided in the next few paragraphs. A complete list of the archetypes, including the category each belongs to and the details of modifications carried out is provided in Appendix F.

Elimination of FuGE base classes

The two base classes from which all other FuGE objects descend, i.e. “Describable” and “Identifiable” are eliminated because their scope is too generic and do not model a specific domain-level concept. “Describable” contains generic placeholders (such as “PropertySets” of type “NameValueType” and annotations of type “OntologyTerm”) and other generic (non-domain specific) information, such as a “Description”, a “URI”, “Security” and “Audit” information. Similarly, the candidate archetype corresponding to the FuGE “Identifiable” class which provides a name and identifier attribute and associations with a “BibliographicReference” and “DatabaseEntry” class are also too generic and not domain specific and are thus eliminated.

Elimination of candidate archetypes that are too generic

The scope of several candidate archetypes was too generic. For example

- CLUSTER.parameter (“A Parameter is a replaceable value in a Parameterizable class”)
- CLUSTER.parameter_value (“The value of a Parameter”)

- CLUSTER.parameterizable (“The types of inputs this Protocol can take e.g. organisms, solutions, data formats etc.”)

do not make sense as stand-alone archetypes. These items would generally be described as ADL structures with an archetype definition and are therefore removed.

Elimination of candidate archetypes describing inbuilt data types

Some of the candidate archetypes described generic data types. For example, objects of the Measurement package define elements that act as wrappers of simple data types, such as “CLUSTER.boolean_value”, which describes a boolean value. These candidate archetypes can be omitted as archetypes because the information they describe is part of the *openEHR* datatype reference model. Thus, the information contained in the automatically created candidate archetype “CLUSTER.boolean_value” would be modelled with the *openEHR* data type DV_BOOLEAN within an archetype definition.

Elimination of “CollectionType” candidate archetypes

Similar to the list type elements in mzML, FuGE makes extensive use of container elements that can be used to list object instances, originating from the classes of the FuGE.Collection package, which are eliminated as these are not suitable as archetypes as explained in Section 4.4.5.

Elimination of candidate archetypes for auxiliary information

Several FuGE packages cover non-domain specific, but auxiliary information, such as support for audit, security and use of ontologies. The correspondingly created candidate archetypes for classes of these packages can thus be eliminated, because they do not represent domain specific concepts, and because the function that the original FuGE classes fulfil is covered on the level of the *openEHR* reference model specification, rather than on the domain-modelling, archetype level. For example, in an *openEHR* archetype-based system, audit information is not defined as part of archetypes definitions, but is dealt with outside the scope of the archetype itself, as described in the *openEHR* Common Information Model [241]. Similarly, security information is not part of an archetype definition [39], but dealt with outside archetypes. All the archetypes corresponding to objects of the Ontology Package are also omitted, since terminology terms are incorporated into archetypes through a terminology binding mechanism as described above.

Candidate archetypes whose scope covers technology specific support information, such as CLUSTER.generic.internal_data (“Data that is internal to an instance XML document stored in an array with a specified encoding. The array will typically use

pointer arithmetic to access values based on the rank (e.g. number of) Dimensions and their respective sizes (e.g. the number of contained DimensionElements)”) are eliminated, since they do not represent a domain-level concept.

Elimination of overlapping archetypes

As for mzML, overlaps between several FuGE archetypes and existing *openEHR* archetypes were detected, for example the information modelled in the automatically created CLUSTER.organization archetype (archetype description: “Organizations are entities like companies, universities, government agencies for which the attributes are self describing”) is already covered by openEHR-EHR-CLUSTER.organisation. Similarly, the information modelled by the FuGE-based CLUSTER.person candidate archetype (modelling “A person for which the attributes are self describing”) is already covered by *openEHR*-EHR-CLUSTER.individual.personal.v1 etc. Thus, when the overlap was complete, i.e. the existing archetypes covered all the information that the automatically created candidate archetypes covered, then the automatically created candidate archetypes are simply eliminated.

Elimination of abstract type candidate archetypes and consolidation of archetypes representing similar sub-classes

In contrast to mzML, FuGE makes a distinction between abstract and non-abstract elements, reflecting abstract and non-abstract UML classes on which the schema file is based. When the “abstract” attribute is set to true, this indicates that there cannot be an instance of that element itself in an XML instance file. Elements that have their abstract attribute set to false, can be instantiated in an instance document. The “substitutionGroup” attribute in the element definition defines which (possibly abstract) element this element can replace in an instance document. This relationship is similar to having a generic archetype that can have specialisations, except that the concept of abstract and non-abstract does not exist in the archetype reference model, so that there cannot be abstract archetypes. Thus, candidate archetypes created for the xsd:complexType elements representing abstract classes are eliminated when equivalent candidate archetypes existed for xsd:complexType element representing non-abstract sub-classes or the same concept. For example, the candidate archetype created for the abstract class Software (CLUSTER.software) and the candidate archetype created for the non-abstract class GenericSoftware (CLUSTER.generic_software) described the same information.

Similarly, in some cases several subclasses of an abstract class existed which modelled distinct parts of the same concept. In these cases, the corresponding candidate

archetypes were unified into one candidate archetype, modelling the multiple distinct aspects. For example, the content of `CLUSTER.generic_material_measurement` (representing “A subclass of `MaterialMeasurement` to be used without being extended in conjunction with `GenericProtocolApplication` and `GenericMaterial` to model measured sources of materials”) content was combined with the `CLUSTER.generic_material` candidate archetype (representing “A subclass of the abstract `Material` class, which should be used in conjunction with controlled vocabulary terms to describe `Materials` of any types used in an investigation”) into a `CLUSTER.generic_material` archetype.

List of remaining FuGE-based candidate archetypes for detailed review

After eliminating all unsuitable candidate archetypes, only 16 candidate archetypes were left that modelled domain-level concepts, listed in Table 4.11.

These remaining candidate archetypes were first updated according to the elimination of other candidate archetypes before reviewing them in detail. For example, some candidate archetypes contained slots for eliminated archetypes, so the slots were either eliminated if they were deemed unnecessary, or modified into an internal ADL structure, as appropriate. For example, in the `CLUSTER.higher_level_analysis` candidate archetype, the slot for the eliminated `CLUSTER.description` candidate archetype is modified to an `ELEMENT` of data type `DV_TEXT`. Further, the leading underscore (“_”) that some element names included is omitted in the naming of corresponding archetype items.

Choice of root archetype class and dealing with multiple non-composite associations

Similar to the reasoning provided above for choosing the correct archetype class as root class for describing a mass spectrometry experiment, it was decided that for describing a functional genomics experiment as a whole, the most appropriate archetype level is the `ENTRY` level and of the five available sub-classes, it was decided that the `OBSERVATION` class is ontologically the most appropriate class to describe the execution and results of an experiment, similar to existing `OBSERVATION.laboratory` type archetypes.

The detailed review of the remaining archetypes, specifically of `CLUSTER.generic_protocol`, `CLUSTER.generic_action`, `CLUSTER.generic_protocol_application` and `CLUSTER.action_application` proved more challenging due to non-hierarchical relationships (non-composite/non-containment associations or aggregations) of `xsd:complexType` elements representing associations of FuGE UML classes in the source (XSD) format that were emulated in the candidate archetypes.

Candidate archetype	Description
CLUSTER.action_application	ActionApplication mirrors the structure of Actions within Protocols and allows ActionDeviations to be reported. RULE: If an Action references a childProtocol, an ActionApplication must be created and reference a childProtocolApplication of the corresponding type (or GenericProtocolApplication) to mirror the Protocol structure. ActionApplications can be (but need not be) created for simple Actions.
CLUSTER.bibliographic_reference	Represents bibliographic references, including the most common attributes. Note that because a BibliographicReference is Identifiable, a DatabaseEntry can also be specified.
CLUSTER.data	Data can be an input to or an output from a ProtocolApplication. Data may be produced from a Material (data acquisition) or from another Data object (data transformation). Examples of Data are gene expression measurements, or phenotypes associated with genetic manipulations.
CLUSTER.database	An address to a repository.
CLUSTER.database_reference	A reference to a record in a database.
CLUSTER.data_partition	An abstract class to represents a segmentation of the data by indexing existing DimensionElements of a Data. The class can be extended to reference specific subclasses of DimensionElement.
CLUSTER.fuge	The root container for accessing all objects.
CLUSTER.generic_action	A GenericAction represents a step within a GenericProtocol. It allows a reference to a sub-GenericProtocol, user entered text to describe the GenericAction or a term from a controlled vocabulary to be given.
CLUSTER.generic_material	A subclass of the abstract Material class, which should be used in conjunction with controlled vocabulary terms to describe Materials of any types used in an investigation.
CLUSTER.generic_protocol	GenericProtocol should be used with GenericAction, GenericParameter or protocolText to describe protocols in a data format for cases where no explicit extension of Protocol has been developed.
CLUSTER.generic_protocol_application	A subclass of ProtocolApplication to be used as it is in data formats without being extended.
CLUSTER.higher_level_analysis	A container for domain-specific analyses performed on data. Subclasses must be created for different domains.
CLUSTER.investigation	An Investigation is a self-contained unit of study that contains one or more types of 'omics technology.
CLUSTER.investigation_component	An overview of the design using a single omics technique.
CLUSTER.sequence	A Sequence is a representation of a DNA, RNA, or protein sequence.
CLUSTER.sequence_annotation_set	This is used to annotate a set of sequences that share the same species, type and PolymerType.

Table 4.11: List of remaining FuGE-based candidate archetypes

In mzML, any non-hierarchical relationships (specifically references to elements instantiated under the list type elements) were easily disentangled and referenced items were included directly in a meaningful context, so that a clear archetype hierarchy could be created with the resulting archetypes without losing context. However, the situation is more complicated in FuGE. In FuGE, non-containment associations are used to give instance data a meaningful context. Some of the relationships in the FuGE schema cannot be as easily disentangled and converted into a true hierarchy needed for the archetype context and a different strategy is needed.

This particularly concerns the distinction made in FuGE between a protocol (modelled with the “Protocol” class) and the application of this protocol at run-time, as represented by the “ProtocolApplication” class: FuGE distinguishes between a protocol, which may be defined once, such as a Standard Operating Procedure (SOP) or a method in a lab book, and its application (modelled with “ProtocolApplication”), which may be carried out many times and can capture runtime values, such as the person who performed the protocol and the date it was carried out [129, 138]. Thus, an experimental protocol can be recorded with the “Protocol” class (e.g. “GenericProtocol”), which consists of a set of steps (“Actions”, such as “GenericAction”). “Actions” themselves can refer to child “Protocols”. Thus, complex laboratory procedures can be represented by nesting “Protocols” and “Actions” that refer to child “Protocols” recursively [138]. In the XML instance file that complies with FuGE, the nesting hierarchy is modelled through referencing of listed protocols. An instance of a (nested) “Protocol” represents the ideal case, for example a SOP, defining the actions that should be carried out as part of this protocol. The runtime contextualization of the defined protocol in time and relating it to material used and the person who carries it out is mirrored by a ProtocolApplication instance (e.g. GenericProtocolApplication). In addition, in an instance of FuGE, each “ProtocolApplication” is linked back to the particular “Protocol” that was applied [129, 138]. A instance of “ProtocolApplication” allows for the description of inputs and outputs of the protocol, such as specific “Material” and “Data” which can be exploited to track samples and data through experimental workflows and data-processing pipelines, for example in a series of data transformations to link the original samples to the processed results (e.g. gene expression matrices) from the original sample, to the processing of the sample, acquisition of raw data and analysis of the created data [129, 138]. The “ProtocolApplication” class also allows for the annotation of deviations from the ideal protocol at runtime. An instance of “ProtocolApplication” contains instances of a “ActionApplications”, mirroring the structure of “Actions” within “Protocols” and allowing “ActionDeviations” to be reported. Thus, for each instance of “Protocol”, a “ProtocolApplication” is instantiated and for each “Action” instance, there is a corresponding instance of “ActionApplication”. These

relationships are shown in the UML models in Figures 4.10 to 4.13:

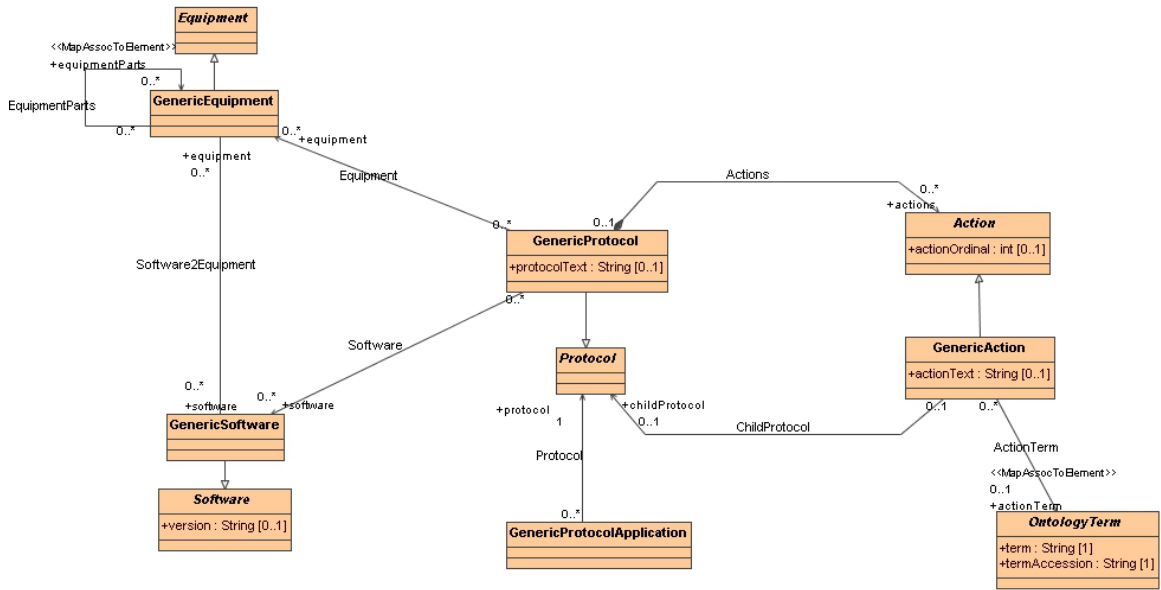


Figure 4.10: GenericProtocol, GenericEquipment, GenericSoftware and GenericAction allow experimental procedures to be represented in FuGE without extension using free text and ontology terms, Source: [223]

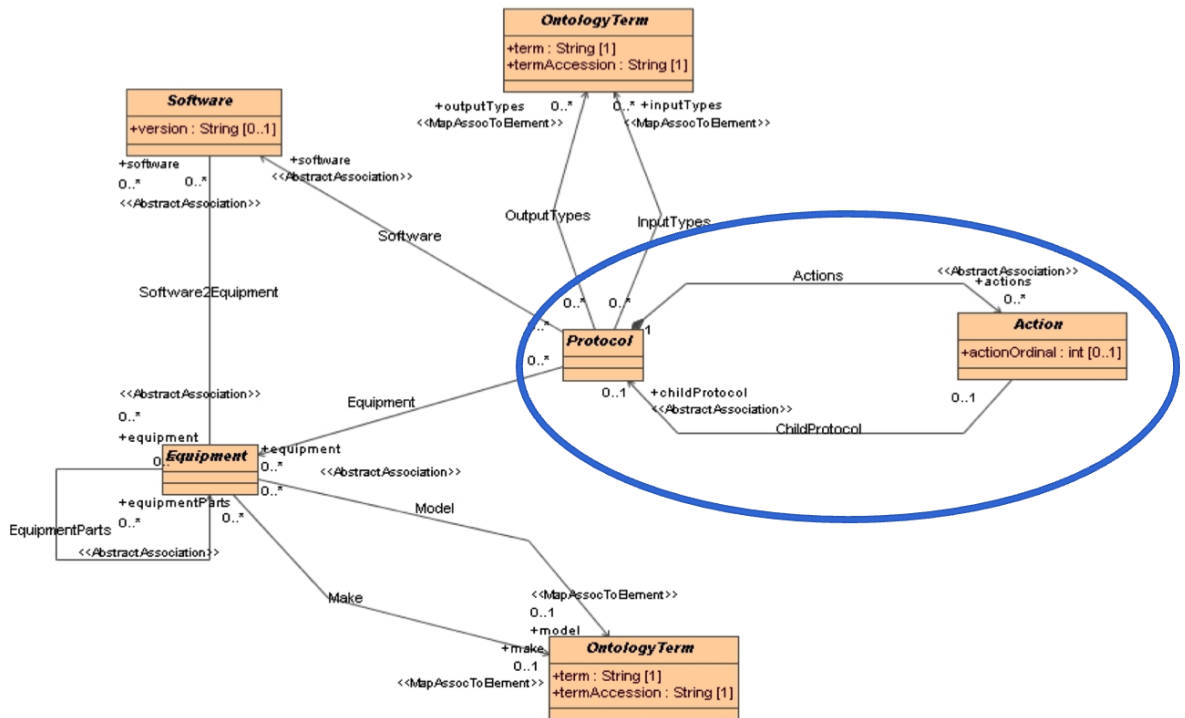


Figure 4.11: The Protocol package contains the abstract classes Protocol, Software, Equipment and Action that should be extended in the development of new modules (Source: [223]). The blue oval highlights the relationships between Protocol and Action.

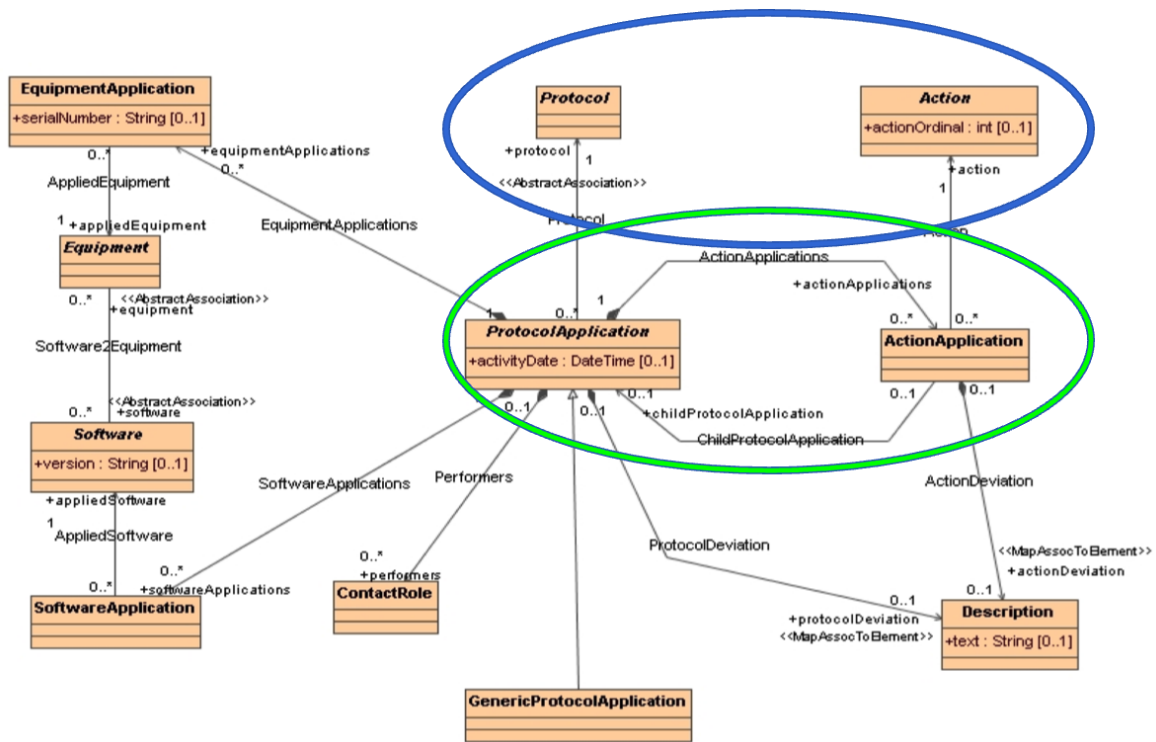


Figure 4.12: ProtocolApplication represents the running of a protocol. ProtocolApplication can record the operator(s) (Performers), the date, and any deviations in the Protocol or Action objects contained within the Protocol (Source: [223]). The relationships between ProtocolApplication and ActionApplication are marked with the green oval, whereas the blue oval corresponds to the one in Figure 4.11 (the relationships between Protocols and Actions are not shown).

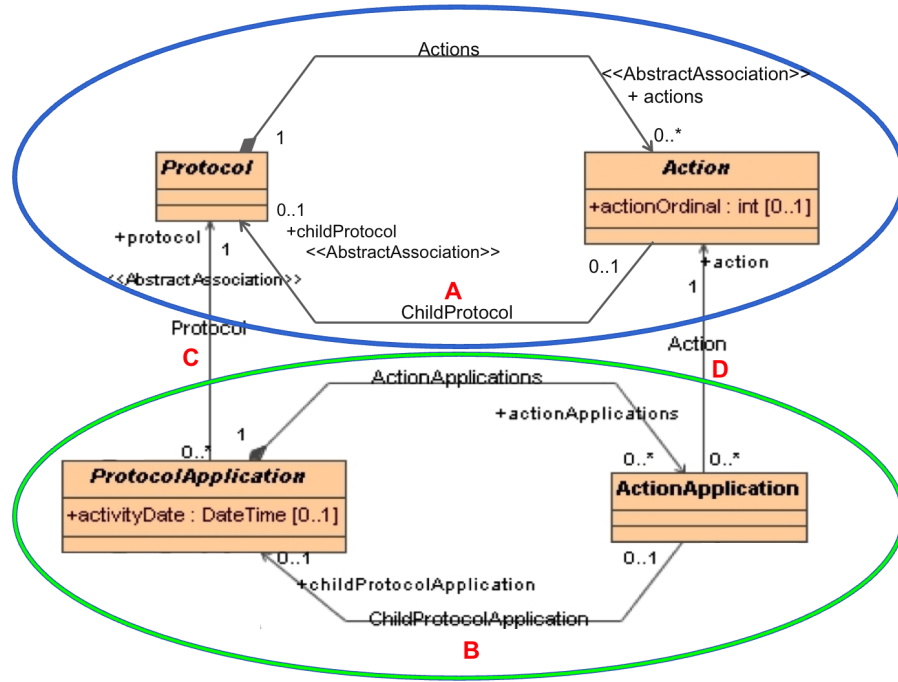


Figure 4.13: This figure is an overlay of Figure 4.11 and Figure 4.12, focussing only on the relationships between Protocols, Actions, ProtocolApplications and ActionApplications, while omitting all other relationships. The blue oval highlights the relationships between Protocols and Actions, and corresponds to the oval in Figure 4.11, and the green oval highlights the relationships between ProtocolApplications and ActionApplications, and corresponds to the oval in Figure 4.12 (adapted from: [223]). Non-containment associations between FuGE classes are labelled with letters A-D (C' and D' between child instances of relevant classes).

In the automatic transformation process to create candidate archetypes from the FuGE XSD file, composite or containment associations were modelled by creating a slot in the archetype to include the referred-to item through the slot mechanism directly, in order to recreate the informational hierarchy directly in the data recording structures.

As a first step, the `CLUSTER.generic_action` archetype was absorbed into the parent `CLUSTER.generic_protocol` archetype, since it is regarded as belonging to the same concept of an experimental protocol. Similarly, the `CLUSTER.action_application` was absorbed into the parent `CLUSTER.generic_protocol_application` archetype. This result of this step is shown graphically in Figure 4.14 (omitting the 'generic' prefix).

The *openEHR* reference model, while allowing for cross-referencing, promotes a more hierarchical organisation of data. Thus, the virtual hierarchies that may be created through referencing (to represent non-composite associations) can be recreated by changing the `DV_EHR_URI` to be used to reference child protocols with a slot for the `CLUSTER.generic_protocol`, so that nested protocols can be modelled by nesting the child protocol in location, rather than by reference.

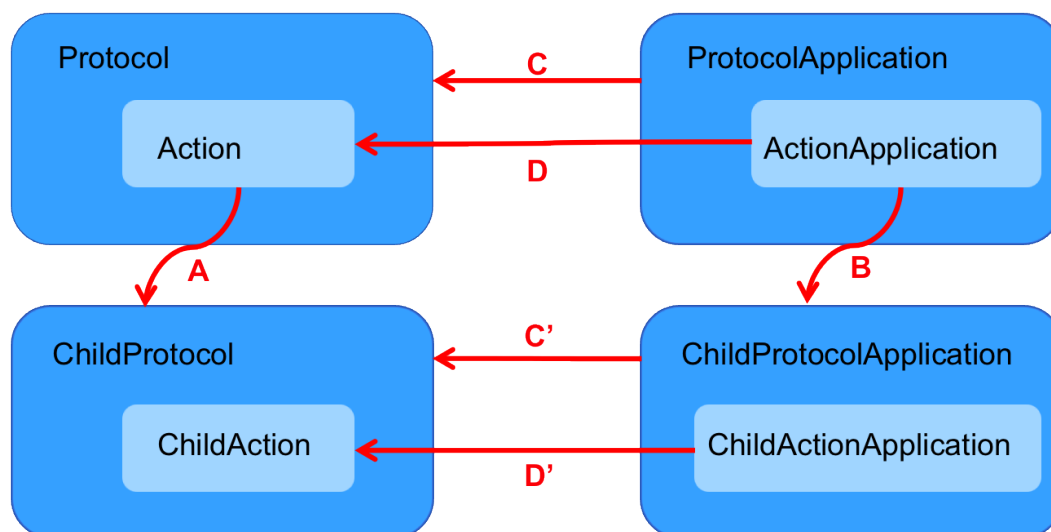


Figure 4.14: Creating archetypes for related FuGE classes: Absorbing Action into Protocol and absorbing ActionApplication into ProtocolApplication to model the containment associations between Protocol and Action and between ProtocolApplication and ActionApplication. Letters A-D label non-containment associations between the FuGE classes on which the development of archetypes is based and correspond to those in Figure 4.13. C' and D' denote relationships between child instances of FuGE classes as labelled.

This is similar to how devices and sub-devices are modelled in the *openEHR* CLUSTER.device archetype, which contains a slot for CLUSTER.device in order to model the sub-components of a device.

In a similar manner, a hierarchy of CLUSTER.generic_protocol_applications can be constructed by changing the DV_EHR_URI reference type under the now internalized CLUSTER.action_application archetype to a slot for a child CLUSTER.generic_protocol_application. The result of this step is illustrated in Figure 4.15 (omitting the 'generic' prefix).

It is possible to link both hierarchies in *openEHR* archetypes through cross-referencing using ELEMENTs with value type DV_EHR_URI in the CLUSTER.generic_protocol_application archetype to refer to the CLUSTER.generic_protocol instance and the internal CLUSTER.action_application instances. However, the *openEHR* reference model promotes a more hierarchical structure, although it does allow cross-referencing of items within records (through DV_EHR_URI), which allows a deviation from hierarchical relationships. Also, the existing CLUSTER.citation archetype could be used to reference information held elsewhere, in the same EHR or external to the EHR. Either alternative would alleviate this situation, and could be used to exactly replicate the complex relationship between Protocols, Actions, sub-protocols and their Application type counterparts across CLUSTER type archetypes. The question is if this separation can (or even should) be upheld in an archetype context and how it could be

represented in high-quality, reusable archetypes that can be used to record the information represented by the concerned FuGE UML classes. The use of cross-referencing is common in XML, but it is discouraged in archetypes, which promote a more strict hierarchical approach. Thus, a solution of how to deal with the two parallel, but highly interlinked FuGE hierarchies of “Protocols” containing “Actions” containing further child-“Protocols” and “ProtocolApplication” containing “ActionApplications” containing further child-“ActionApplications” still needs to be found.

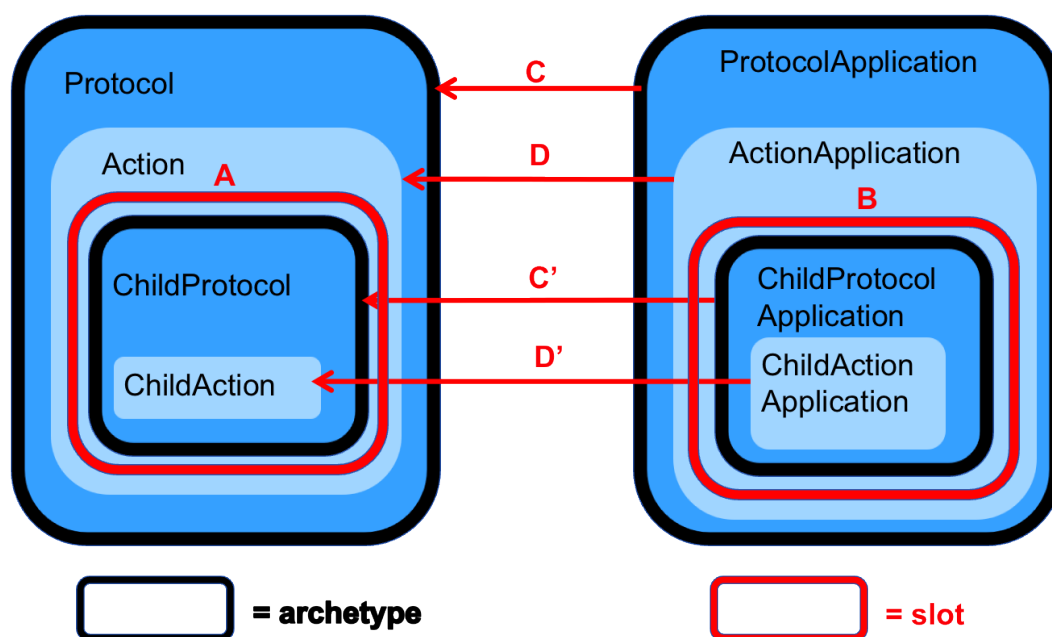


Figure 4.15: Creating archetypes for related FuGE classes: Creating slots for Child-Protocols and ChildProtocolApplications, corresponding to the non-containment associations labelled A and B in Figure 4.14. Labels C, D, C' and D' correspond to those in Figure 4.14, denoting remaining non-containment associations.

Working towards finding a solution, it should be noted that the OBSERVATION class, like all the other sub-classes of the abstract *openEHR* CARE_ENTRY class contains an attribute called “protocol” which can be used to describe the method used to produce the information in that particular entry [42]. Thus, since the aim is not to replicate the structures and relationships of FuGE classes exactly in archetypes, but to create reusable archetypes that can be used to record functional genomics experiments, it was decided to combine both the modelling of a “Protocol” containing “Actions” and the application of a protocol (“ProtocolApplication”), containing “ActionApplications” that mirror the corresponding “Protocol” and “Actions” into one CLUSTER archetype that contains a slot for itself, thus allowing for nesting of protocols, included under the “protocol” attribute in an OBSERVATION archetype. The result of this step is illustrated in Figure 4.16 (omitting the ‘generic’ prefix). All non-containment associations have thus been eliminated between Protocol, Action, ProtocolApplication

and ActionApplication (and their child instances). This adheres more closely to the hierarchical nature of archetypes by reducing the amount of cross-referencing without losing context.

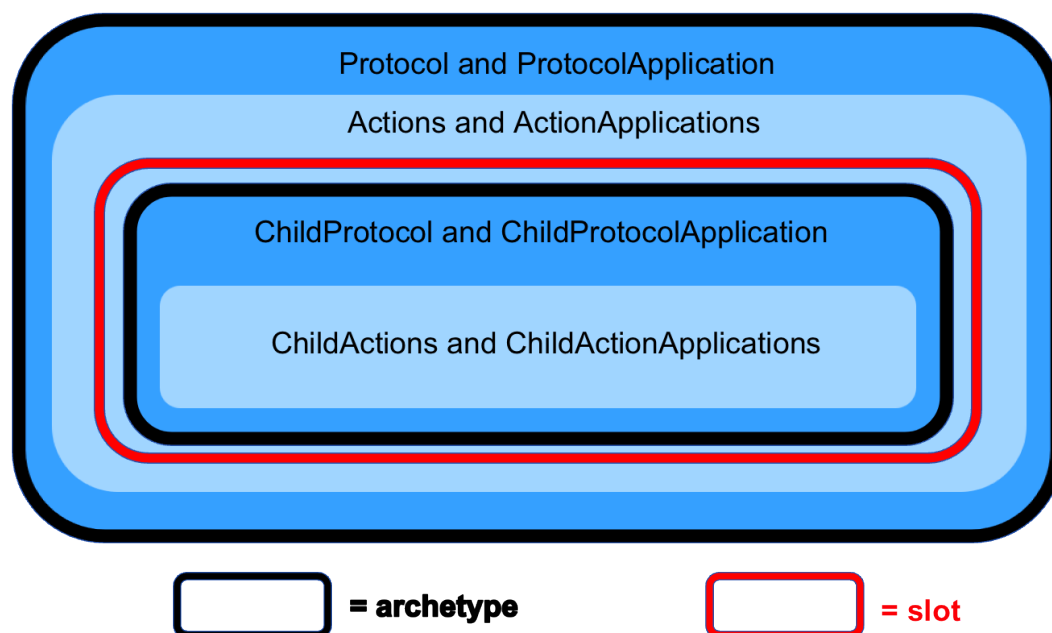


Figure 4.16: Creating archetypes for related FuGE classes: Combining Protocols containing Actions with ProtocolApplications containing ActionApplications into one archetype.

The remaining archetypes were re-organised, with the OBSERVATION.omic_experiment at the top of the hierarchy. Information related to the experimental workflow is located under the “protocol” attribute of the OBSERVATION archetype and information related to the data produced in the experiment is modelled under the “data” attribute in the OBSERVATION archetype. These archetypes, together with existing archetypes, may then be customised in templates to record simple functional genomic experiments.

Final list of FuGE-based functional genomics archetypes

Table 4.12 provides the final list of archetypes for describing a functional genomics experiment. The ADL files representing these archetypes are provided on the CD supplied with this thesis. The final list of archetypes is provided in Table 4.12.

Final archetype	Description
CLUSTER.bibliographic_reference	Represents bibliographic references, including the most common attributes. Note that because a BibliographicReference is Identifiable, a DatabaseEntry can also be specified.
CLUSTER.biological_molecule_sequence	A Sequence is a representation of a DNA, RNA, or protein sequence.

CLUSTER.database	Description of a repository.
CLUSTER.database_reference	A reference to a record in a database.
CLUSTER.material	To describe Materials of any types used in an investigation.
CLUSTER.protocol	A generic experimental protocol that can be nested to describe complex experimental protocols. For cases where no explicit extension of Protocol has been developed.
EVALUATION.omic_investigation	To describe the overall outcome of an omic investigation.
OBSERVATION.omic_experiment	To describe a single omic experiment.

Table 4.12: List of final functional genomics archetypes

4.5.5 Summary of the creation of archetypes from FuGE

From the initially large amount of candidate archetypes, only a small number of domain-specific reusable archetypes remained in the end. This is due to the fact that the FuGE data format is based on a generic object model that provides a generic framework for recording functional genomic experiments, and thus does not provide much archetypeable domain-specific information itself. In general, FuGE classes are expected to be extended to describe more domain specific procedures by placing more specific constraints on the details captured. In this regard, the FuGE specification can be considered to be closer ontologically to the level of the *openEHR* reference model, rather than modelling domain specific information. It specifies the generic classes and their relationships that can be used to construct domain specific models of information (within the omic domain), rather than providing details about recording information in a specific omic domain, whereas the *openEHR* reference model specifies classes and their relationships that can be used to construct domain specific models of information within the healthcare domain. Thus, the value of the archetypes developed from FuGE are restricted to high-level description of simple omic experiments.

In addition to the generic FuGE object model, a number of domain-specific extensions for FuGE exist and further ones are planned or are being developed, as discussed in Chapter 2. One such domain-specific extension of the generic FuGE object model is GelML. Since GelML models domain-specific information in a generic manner, it was thought that it would provide more archetypeable information structures than the generic FuGE object model. The process and results of applying the semi-automatic process developed above was thus applied to GelML with the intention to create omic archetypes based on GelML. This process is described in the next sections.

4.5.6 The GelML data exchange format

Gel electrophoresis (GE) is a specific omic technique used for the separation of proteins from complex mixtures in a polyacrylamide gel matrix by molecular weight (one-dimensional electrophoresis) or by isoelectric point (pI) and molecular weight (two-dimensional electrophoresis) [242]. To visualise proteins on an electrophoresed gel, they are generally labeled or stained [201]. An image of an electrophoresed gel following 2-dimensional electrophoresis is shown in Figure 4.17. Next, image analysis is performed to assign protein identities or carry out quantitation [201]. GelML can be used to record both the data (for example gel images) and metadata (such as experimental protocols) from gel electrophoresis experiments and can be used to create MIAPE-compliant documents [201]. GelML can be used in conjunction with other formats in a modular manner [201].

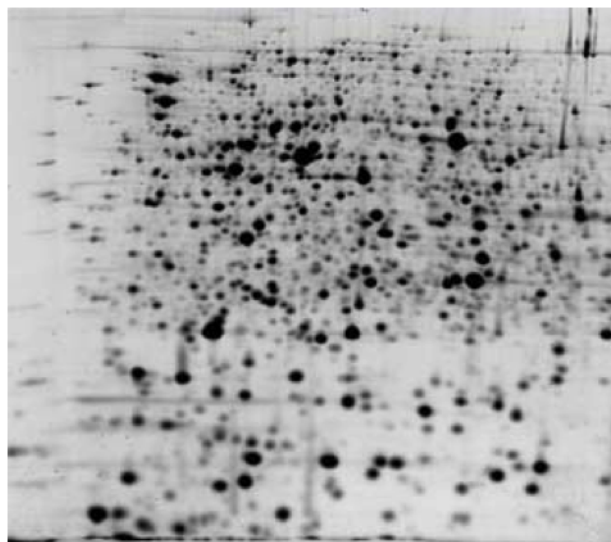


Figure 4.17: Image of a stained gel following 2-D electrophoresis.

The gel electrophoresis markup language (GelML) is the data exchange format to record gel electrophoresis experiments in proteomics, developed by the HUPO PSI Gel working group as an extension to the FuGE data model [203, 201]. GelML covers the application of gel electrophoresis protocols in a proteomics experiment, direct and indirect (e.g. through Western blotting) protein detection and acquisition of the image from the gel matrices [201]. GelML does not cover sample preparation prior to electrophoresis (this may be recorded using FuGE) and does not cover analysis of the image or identification of the proteins (this may be captured with MS or with formats mzML [21] or mzIdentML¹⁸ [201]).

PSI has also developed an accompanying CV, sepCV (Sample Processing and Separation techniques) to standardise vocabulary used in peptide and protein separation

¹⁸mzIdentML, <http://www.psidev.info/mzidentml>

experiments [201].

A graphical representation of the components of an example GelML file is provided in Figure 4.18.

Example GelML contents

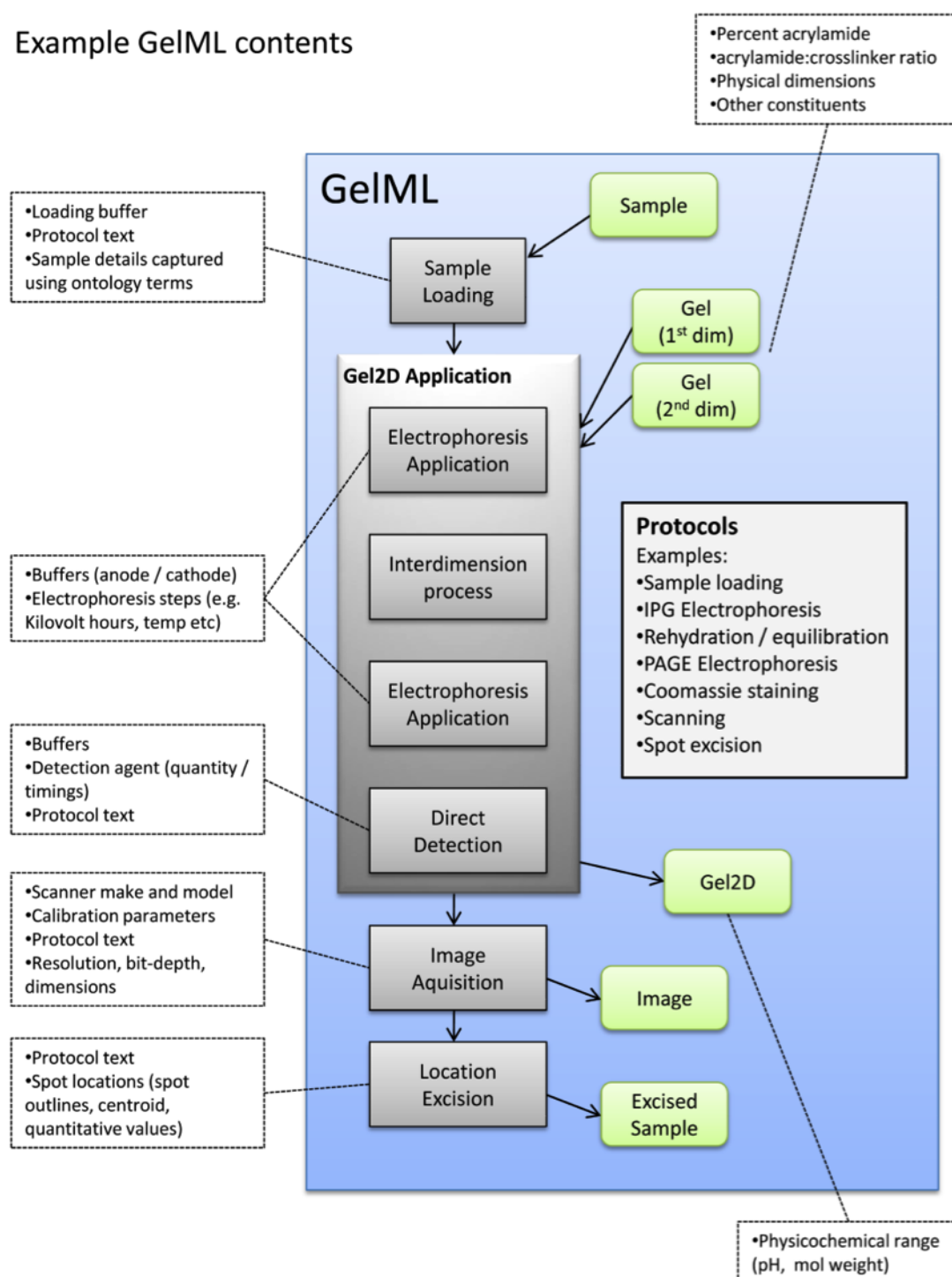


Figure 4.18: A graphical representation of example components from a GelML file, and certain key details that should be captured in each section. Standard rectangles indicate ProtocolApplications, rounded rectangles indicate Materials or Data. Source: [201]

In detail, GelML is organised into the following sub-sections for each step in a GE experiment: gel materials, optionally the manufacture of the gel; one-dimensional gel electrophoresis; two dimensional gel electrophoresis; other, “non-standard” methods of gel electrophoresis (such as 3-dimensional geometry gel electrophoresis); sample loading; electrophoresis; protein detection; image acquisition and the excision of locations on gels [201].

4.5.7 Syntactic transformation of GelML

The GelML documentation and format and auxiliary files are available on the HUPO PSI code repository¹⁹.

The automatic transformation software that was previously adapted to the structure of FuGE was minimally modified to accommodate for the fact that GelML is an extension of FuGE ML. For example, many of the GelML element definitions are extensions of FuGE elements, so in order to create archetypes from the GelML schema, both the FuGE schema and the GelML schema files need to be parsed. GelML also reuses many of the FuGE elements directly and without modification. Furthermore, in contrast to the generic FuGE ML, GelML is associated with an ontology mapping file (GelML-mapping.xml) and an ontology referenced in the mapping file (sep.obo), so the transformation software made use of these files in the transformation process, similar to how this has been done for mzML, but adapted for the different ontology term placeholder structure as described for FuGE ML in Section 4.5.3.

GelML defines 76 xsd:complexType elements, which result in the creation of 76 ADL files to act as candidate archetypes for Phase 2 of the omic archetype development process. A list of the candidate archetypes is provided in Appendix G.

4.5.8 Semantic review of candidate archetypes for gel electrophoresis

Similar to FuGE, many of the candidate archetypes could be eliminated before reorganisation, for similar reasons as described in Section 4.5.4 for candidate archetypes created from the FuGE data format.

For example, similar to some FuGE archetypes, many candidate archetypes created from GelML modelled concepts that are too generic or are covered differently in *openEHR*, such as “Collection” type candidate archetypes (CLUSTER.gelML_material_collection, CLUSTER.gelML_data_collection, CLUSTER.gelML_protocol_collection). These are eliminated, and information contained in these lists is included, as before,

¹⁹GelML, <http://code.google.com/p/gelml/>

by replacing a non-composite association with a containment association using the slot mechanism within the archetype that contains the most closely related parent concept.

Others also described inbuilt data types and could be deleted and the inbuilt data types used instead. However, while in FuGE, elements described generic data types, such as *Parameter*, which refers to a *Measurement*, in GelML, these were often extended into more specific elements, such as “*Mass*” or “*Concentration*” (extensions of “*Parameter*”) or “*TimePoint*” (extension of “*TimeParameter*”). Again, in archetypes, these concepts are too generic and generally represented with internal ADL structures, using *openEHR* data types. For example, mass in archetypes can be represented as shown in Listing 4.26.

```

ELEMENT[atxxxx] occurrences matches {0..1} matches {&— Mass
  value matches {
    CDV_QUANTITY <
      property = <[openEHR::124]>
      list = <
        [ " 1" ] = <
          units = <"pg">
          magnitude = <|>=0.0|>
        >
        [ " 2" ] = <
          units = <"ug">
          magnitude = <|>=0.0|>
        >
        [ " 3" ] = <
          units = <"mg">
          magnitude = <|>=0.0|>
        >
        [ " 4" ] = <
          units = <"gm">
          magnitude = <|>=0.0|>
        >
        [ " 5" ] = <
          units = <"kg">
          magnitude = <|>=0.0|>
        >
      >
    >
  }
}

```

Listing 4.26: Representing mass in ADL

Thus, wherever a parent archetype included a mass archetype through the slot mechanism, the slot was replaced with the ADL snippet shown in Listing 4.26. This was carried out similarly for those elements that could be represented with archetype structures, e.g. for *CLUSTER.concentration* and the *CLUSTER.volume* and other candidate archetypes.

Again, the strategy was to eliminate archetypes that are too generic and do not describe domain specific information and the remaining candidate archetypes were then

reviewed in detail. Instead of repeating the reasons for elimination here, Appendix G provides a list of all candidate archetypes created for GelML and details which candidate archetypes are eliminated and the reason for elimination, which refer to any of the reasons explained in the context of FuGE in Section 4.5.4.

In contrast to FuGE-based candidate archetypes, only a small number of GelML-based candidate archetypes were eliminated in this step because their content does not model a domain-level concept.

After the process of elimination, the remaining candidate were reviewed in detail. During this process, many of the modifications carried out and the reasoning for these modifications are similar to those carried out on the remaining FuGE candidate archetypes, and are not re-discussed here. Briefly, many similar actions were carried out as for the remaining FuGE archetypes, such as adjusting slots referring to eliminated archetypes accordingly or creating internal ADL structures as appropriate. The list of candidate archetypes created for GelML in Appendix G provides details of the modifications carried out. Similarly, as in FuGE, many element names started with a leading underscore ("_"), which was omitted in the naming of corresponding archetype items.

Those issues and their resolutions particular to GelML-based candidate archetypes are discussed below.

Problems with GelML CV mapping rules

The use of ontology terms in an instance of GelML is controlled by the GelML-mapping.xml file. However, several inconsistencies were encountered, resulting in problems when trying to map appropriate `OntologyTerm_ref` descriptions to ontology terms. For example, the GelML data format specifies several `OntologyTerm_refs`, which act as placeholders for ontology terms, but the GelML mapping file fails to provide corresponding links to an ontology, so that no `CODE_PHRASEs` for the corresponding `DV_CODED-TEXT` items can be provided. Finally, the mapping file defines rules for elements that do not exist in the GelML specification ("orphan mapping rules"). An attempt was made to incorporate the information provided by the mapped ontology terms where possible.

Eliminating overlap in scope of candidate archetypes

Some of the automatically created candidate archetypes overlapped in scope and thus were combined. Some examples are given below: For example, the candidate archetypes for `CLUSTER.gel_2d`, `CLUSTER.gel_1d` and `CLUSTER.other_gel` were all combined into one `CLUSTER.electrophoresed_gel` archetype, which can then be renamed ap-

appropriately in parent archetypes that include a slot for these archetypes or alternatively these can be renamed at template level. In addition, CLUSTER.circle, CLUSTER.point, CLUSTER.rectangle, CLUSTER.boundary_chain and CLUSTER.boundary_chain_set were absorbed into the CLUSTER.feature_location archetype. Then, CLUSTER.feature_location, CLUSTER.gel_location and CLUSTER.location_measure and CLUSTER.location_set were combined into a CLUSTER.feature_location archetype.

Overlap was also detected in the candidate archetypes CLUSTER.gel_1d_application, CLUSTER.gel_2d_application and CLUSTER.other_gel_application, which all describe different types of a Gel Electrophoresis. Similarly, CLUSTER.gel_experiment overlaps with CLUSTER.other_gel_experiment, CLUSTER.gel_2d_experiment and CLUSTER.gel_1d_experiment. Again, each group of these related archetypes are unified into one archetype, resulting in one combined CLUSTER.gel_experiment and CLUSTER.gel_application archetype.

Choice of root archetype class and dealing with multiple non-composite associations

As for mzML and FuGE, the root archetype class for GelML was chosen to be an OBSERVATION class ENTRY level archetype, since the experiment results in an OBSERVATION, in this particular case in an image of an electrophoresed gel. Thus, the OBSERVATION.gel_electrophoresis_experiment archetype forms the root archetype for a gel electrophoresis experiment, based on the CLUSTER.gel_experiment candidate archetype which combined all four CLUSTER.X_gel_X_experiment archetypes as discussed in the previous section. In the OBSERVATION.gel_electrophoresis_experiment archetype, the experimental protocol is modelled under the “protocol” attribute of the OBSERVATION archetype and the data produced in this experiment, represented by a description and image produced from the electrophoresed gel is modelled under the “data” attribute of the OBSERVATION archetype.

Since GelML is based on FuGE, many of the candidate archetypes created for GelML are extensions to FuGE classes. This means conflicts encountered in the creation of candidate archetypes from particular FuGE classes are re-encountered in candidate archetypes created for extensions to these FuGE classes. This in particular concerns the issue of non-composite associations between the FuGE classes “Protocol”, “Action”, “ProtocolApplication” and “ActionApplication”. The same solution was applied to these candidate archetypes, i.e. candidate archetypes modelling “Actions” were absorbed into the parent “Protocol” candidate archetype and candidate archetype modelling “ActionApplications” were absorbed into the parent “ActionApplication” candidate archetypes. Then, as for the two related FuGE candidate archetype classes for “Protocol” and “ProtocolApplication”, for each of the two candidate archetypes

representing the extensions of these classes, these were combined into one candidate archetype. For example, the candidate archetype for “SampleLoadingProtocol” was combined with the candidate archetype for “SampleLoadingApplication”, after child “Actions” and child “ActionApplications” were absorbed into each parent candidate archetype respectively.

Reusability considerations

Another aim of the review process is to create reusable archetypes, i.e. to reduce specificity in archetypes to maximise reusability of the resulting archetypes, where possible. Thus, candidate archetypes that only seem useful to record information within the context of a gel experiment are thus absorbed into the OBSERVATION.gel_electrophoresis_experiment, rather than modelled as stand-alone archetypes. This for example concerns the combined CLUSTER.gel_application archetype modelling the application of a gel electrophoresis protocol, which is not expected to be useful outside the context of an electrophoresis experiment. Therefore, the content of the CLUSTER.gel_application archetype into the “protocol” section of the parent OBSERVATION.gel_electrophoresis_experiment archetype.

Similarly, when archetypes seemed generally applicable, but contained descriptions of gel specific terms, these were evaluated to see if the terms could be changed into more generic terms, so that the archetype may be reused in other contexts and not just in the gel electrophoresis context. This was for example applied to the CLUSTER.gel_manufacture archetype, used to describe the manufacture of a gel used for electrophoresis, which could be generalised into a generic CLUSTER.material_manufacture archetype by replacing “gel” with “material” in the archetype, so that it could be reused in a different context to describe the manufacture of any material used, thus making it reusable in a larger context. Another example is the CLUSTER.detection_application candidate archetype that was “generalised” so that it can be reused in other experiments in which a staining and detection protocol is carried out, such as in situ hybridisation and immunohistochemistry experiments as required by the MISFISHIE guidelines [149].

Detailed content review of remaining candidate archetypes for GelML

As before, descriptions, names etc. were modified in some cases to more accurately describe the information they contain. In the remaining candidate archetypes, the ADL structures created for Describable and Identifiable were eliminated, as in FuGE. Upon inspection of the provided GelML instance files, it was noticed that in a lot of cases the name attribute is used to provide an extensive human readable description,

rather than a name. Thus, where this made sense, the archetypes have been extended to include a DV_TEXT item of name “description” since it is more appropriate than using a DV_TEXT element of name “name”. Furthermore, it was noticed, that the “id” attribute was used entirely for internal referencing in the XML instance file and thus unnecessary in the archetype structure. Only when an “id” attribute also has a “real-life” equivalent or relevance (e.g. representing the identifier of a sample used to identify the sample outside the virtual environment), then an item to hold this information (either as DV_TEXT or DV_IDENTIFIER) is included.

Further, due to the XML extension mechanism, extended elements automatically inherit all the data items from the parent element (including those of Describable and Identifiable), which in a lot of cases are not actually going to be used in the instance files. Due to the way the transformation software was written, corresponding candidate archetypes automatically include every item inherited from parent objects, for example, any extension to the FuGE:ProtocolApplication object automatically contains slots for FuGE:ParameterValue, FuGE:ActionApplication, FuGE:SoftwareApplication, FuGE:EquipmentApplication etc. In many cases these items are unnecessary, since they may never be used, for example there is no need for a CLUSTER.software_application slot in CLUSTER.gel_1d_application, CLUSTER.gel_2d_application, CLUSTER.gel_manufacture_application etc., so items like these were deleted.

Final list of GelML-based archetypes

The number of initial candidate archetypes is thus reduced to 12 final archetypes that can be customised (together with existing archetypes) into a template to record the specifics of any gel electrophoresis experiment. Table 4.13 provides the final list of archetypes for describing a gel electrophoresis experiment. The ADL files representing these archetypes are provided on the CD supplied with this thesis.

Final archetype	Description
CLUSTER.electrophoresed_gel.v1	For describing gels following electrophoresis, e.g. a gel after applying 1-D or 2-D or other types of electrophoresis to separate proteins or other materials on a gel substrate. E.g. a 2-D gel is the material produced by applying electrophoresis to the combination of the two gel substrates and the proteins that have been separated. A 1-D gel is the material produced from a one-dimensional electrophoresis applied to the combination of the gel substrate with the proteins that have been separated.
CLUSTER.electrophoresis.v1	An application of an electrophoresis protocol, capturing the details of electrophoresis.
CLUSTER.feature_location.v1	To describe a location on a material.

CLUSTER.gel.v1	The gel matrix used in any dimension of an electrophoresis assay.
CLUSTER.image_acquisition.v1	An application of an image acquisition protocol describing the scanning or photography of a material to produce an image.
CLUSTER.image.v1	A description of an image.
CLUSTER.manufacture.v1	To describe the manufacture of an object or a material.
CLUSTER.material_excision.v1	The application of a protocol that describes the excision of locations on a material.
CLUSTER.molecule_detection.v1	To describe the process by which molecules are to be detected on a material, by a direct process e.g. staining. If two stainings are done, the destaining should be the first stage of the second staining process.
CLUSTER.sample_loading.v1	To describe how samples are loaded onto a material.
CLUSTER.substance_mixture.v1	To describe the creation of a mixture of substances, for example to be used for specifying the components of buffers, solutions etc.
OBSERVATION.gel_electrophoresis_experiment.v1	To describe the execution and result of a gel electrophoresis experiment.

Table 4.13: List of final gel electrophoresis archetypes

4.5.9 Summary of the creation of archetypes from GelML

The GelML data format, which is an extension of the FuGE format, models much more domain specific information than FuGE, which resulted in a set of reusable omic archetypes that can be composed to record any gel electrophoresis experiment in an interoperable manner. The resulting archetypes generically model the component concepts so that they can be reused in different contexts.

As expected, since GelML is an extension of FuGE, the transformation software adapted for FuGE could be reused with minimal modification for GelML. In addition, many of the issues encountered and decisions for resolution taken during the semantic review process for FuGE applied also to GelML, so that the review process could be carried out in a more efficient manner.

4.6 Summary

The eBMRR needs to be able to store both clinical patient information and the results of experimental analysis on patient biospecimen, such as omic data from high-throughput analysis. Since the implementation of the eBMRR is based on the archetype approach developed by *openEHR*, the clinical information in the eBMRR may largely be recorded with existing archetypes in the archetype-based record, as shown for the

PCRC biobank record described in Chapter 3. This chapter described an approach for developing omic archetypes that can be used to record omic data won from high-throughput analysis on patients' biospecimen, to fulfil the second part of the data recording requirements of the eBMR. The development of archetypes ideally follows a consensus-based process, identifying and agreeing on concepts and specific data items to be recorded belonging to these concepts in a specific domain or sub-domain, specifying and agreeing on the particular meta-information of each item and their organisation and relationships within archetypes.

Instead of starting from scratch, it was decided that the ongoing development of omic data recording standards may be exploited in the development of omic archetypes, since these standards already provide consensus on what information should be recorded. Together with accompanying data formats that have been developed to allow for the sharing of omic information on the level of technology, it was shown that most information needed for the development of omic archetypes could be extracted from these. Furthermore, it was thought that if a mapping could be established between the definition of the data formats, which are often based on XML and thus computationally accessible, and the definition of archetypes, then parts of the process of creating archetypes may be automated. A two-phase process to create omic archetypes from existing data formats is devised. In Phase 1, a mapping between the definition of the data formats and the definition of archetypes is devised and applied to create a set of candidate archetypes. This phase carries the data in the original data formats into the *openEHR* computational context. In Phase 2, the candidate archetypes developed in Phase 1 are reviewed and modified to comply with the semantic and design requirements of archetypes and resolve overlap conflicts.

As a start, the mature and widely used mzML data format, described as an XML schema and auxiliary files used for its instantiation, the CVMappingsRules file and associated ontologies, was investigated and a mapping established between the schema definition and auxiliary files and the archetype definition. This mapping was then used to develop bespoke software that applied these mappings and output a set of candidate archetypes, similar to a set of integration archetypes, which mimicked the structure of an instance mzML file. An advantage of automating the process was to save time over the manual use of an archetype editor and the automatic incorporation of ontology terms mapped in the CVMappingsRules. The candidate archetypes were then reviewed and modified to adhere to semantic archetype design principles, resulting in a set of 'properly' designed archetypes that can be used to record the experimental procedure and results of any mass spectrometry experiment. Common conflicts and their resolution during this process are described.

Then, to test general applicability of this strategy, this process was applied to two

more XML-based data formats: FuGE and the FuGE-extension GelML. FuGE has a special status in that it has been developed to provide a framework for harmonisation amongst other omic data formats. It was found that the software to automatically create archetypes from the underlying XML-based data format developed for mzML could essentially be reused after some modifications were carried out to account for structural differences of element definitions in FuGE(-based) format(s). During the review process, it became clear, however, that most of the candidate archetypes created from FuGE were too generic and covered content that was ontologically more on the level of a modelling specification for omic data formats, rather than providing domain specific information, resulting in a small number of reusable omic archetypes. The application of the process to GelML, an extension of FuGE was also successful, resulting in a set of generic archetypes that can be used to describe the execution and results of a gel electrophoresis experiment.

Thus, it was found that existing omic data formats could be used as basis for the creation of omic archetypes and allow for the automation of part of the process. Basing the development of omic archetypes on existing omic data formats eliminates some of the more time-consuming steps in the archetype development process, such as defining and finding consensus on what information that should be recorded and the meta-information on how specific data items should be recorded. It has been shown that once a mapping is established between the source and target formats, generic transformation software can be developed that can be applied to other, XML-based omic data formats with minimal modifications to create sets of candidate archetypes from other XML-based omic data formats in Phase 1 of the process, leaving Phase 2, the semantic remodelling phase, to become the most time-consuming part. However, lessons learned from the semantic remodelling process of FuGE and GelML could possibly be incorporated into the transformation software in future, to speed up the process of creating semantically sound omic archetypes from further FuGE-based data formats. For example, the software could be modified to automatically create an ELEMENT constraining DV_BOOLEAN for the wrapper element defined in FuGE for the xsd:boolean data type.

The next chapter provides an evaluation of using the *openEHR* archetype approach for the development of an eBMRR for biobanks, discussing both the challenges and feasibility of reusing existing archetypes to model the clinical information in a legacy database of a biobank and the feasibility of modelling omic information, based on standard omic data formats, with the *openEHR* archetype modelling approach.

Chapter 5

Evaluation and Discussion

5.1 Introduction

This thesis set out to investigate whether the *openEHR* approach would be a feasible approach to use for the development of a generic electronic biomedical research record based on the study participant/donor for biobanks that integrates clinical and research information that can be shared amongst collaborating researchers. In order to answer this question, a number of objectives were identified. In particular, one of the main objectives concerned the modelling of the information required for the implementation of the eBMRR with *openEHR* archetypes. This objective was further divided into two particular tasks: the modelling of the clinical information and the modelling of the research information. The modelling of clinical knowledge in clinical archetypes to enable the recording of clinical information in an interoperable manner has been going on for many years in the archetype community to support the development of interoperable EHRs. Thus, instead of creating new archetypes to model the clinical information in the biobank that may jeopardise the interoperability goal because of overlap with existing clinical archetypes, instead, the feasibility of reusing existing clinical archetypes, where possible, to represent the clinical background information about a patient or sample donor in a biobank setting was explored. To do that, an experiment was carried out to evaluate the feasibility of reusing existing clinical archetypes to represent the clinical information in an existing biobank, so that the information in the legacy database could be made available through interoperable archetypes. The research related information, for example omic data and other concepts in research, such as experimental protocols, need to be modelled with a set of generic and customisable archetypes that can be easily adapted to a specific biobank or study. To model omic archetypes, the approach used in this research was to leverage existing data recording standards in the research domain for recording sample handling and experimental results and analysis to design these archetypes. The result is a set of generic and flexible archetypes, which can be adapted to and reused to record specific omic research information. This chapter evaluates the extent to which those objectives have been met within the overall context of

the research question.

Considering the objectives reiterated above, this evaluation is provided in several parts. First, a discussion of the challenges and feasibility of modelling the clinical and related information in an existing biobank, namely the Irish PCRC biobank with existing generic archetypes is provided, including a discussion on the consequences of these challenges on integrating legacy data into an archetype system. Secondly, an evaluation of the development of a set of *openEHR* archetypes that generically model omic research information, based on standard omic data formats, is given. This evaluation also includes a discussion on the 2-phase approach that was developed to facilitate the development of omic archetypes from standard omic data formats, which first carries out an automated syntactic conversion of the data format into archetype format, bringing the meta-data description into the computational realm of archetypes, followed by a review of the candidate archetypes to create semantically sound archetypes. This evaluation will be carried out in the context of the specific data formats used as basis for the development of these archetypes in this research, i.e. concerning the data format for mass spectrometry (mzML), and the data format for electrophoresis (GelML), which is based on the functional genomic experiment data model (FuGE). Finally, the feasibility of basing the information modelling in the proposed eBMRR on a patient-centric system to integrate both clinical and omic research information in a single record in a research environment is discussed.

5.2 Evaluation of modelling the information in the PCRC database with *openEHR* archetypes and templates

This section discusses the challenges encountered in modelling the clinical and related information in a biobank, in the context of the Irish PCRC biobank with existing generic archetypes and the consequences of these challenges on integrating legacy data into an archetype system. An earlier evaluation of these findings has been published in [232]. Here, an updated and more comprehensive account of these findings is presented. An observation regarding the type of the archetypes that have been developed during this research is also provided.

5.2.1 Modelling challenges

While it was possible to fully cover all the fields in the biobank database with new and existing archetypes, the process of modelling the information in the PCRC database

with archetypes posed several challenges and various issues were encountered, which are discussed below. Some of these issues are due to the immaturity of the archetype approach, such as:

- Immaturity of modelling support tools
- Lack of modelling guidelines or best practice rules
- Overlapping archetypes
- Small number of stable archetypes
- Immaturity of the PCRC BIMS database schema

Others are due to the fact that this research attempts to reuse archetypes that were developed for use within an EHR to support patient-care in a different (biomedical) context, which essentially represents secondary use of clinical information, where the information need, although covering the same concepts, may be slightly different. Yet others can be attributed to common legacy database integration issues, such as when the same information is stored in both systems in a way that prevents the establishment of a simple one-to-one mapping between a field in the legacy database and an ELEMENT in an archetype. These issues and their consequences on the feasibility of using an archetype-based approach and on the feasibility of integrating legacy information are discussed below in more detail.

Immaturity of modelling support tools

A number of tools to support archetype development and validation exist. For example, *openEHR* provide a GUI-based archetype editor¹ that comes with an inbuilt Archetype Validator that automatically validates archetypes with regard to correctness of syntax and semantics in the AOM format against the underlying Reference Model. The *openEHR* Archetype Editor can be used as stand-alone or as part of the ADL workbench². The ADL workbench is a multi-model driven GUI application that can be used to edit and review archetypes. Another archetype editor is provided by the Linköping University³.

When this research was started, the *openEHR* Archetype Editor and the Linköping University Archetype Editor were the only two editors freely available. The *openEHR* Archetype Editor has been used extensively during this research, for the development

¹*openEHR* Archetype Editor, http://www.openehr.org/svn/knowledge_tools_dotnet/TRUNK/ArchetypeEditor/Help/index.html

²ADL workbench, http://www.openehr.org/svn/ref_impl_eiffel/TRUNK/apps/adl_workbench/doc/web/index.html

³LiU Archetype Editor, <http://www.imt.liu.se/mi/ehr/tools/>

or modification of archetypes for the PCRC database as described in Chapter 3 and for the validation and modification of omic archetypes as described in Chapter 4. The Linköping University Archetype Editor was initially used in this thesis, but it was found that the Windows based (.NET) *openEHR* Archetype Editor was more mature and reliable than the java-based Linköping University Archetype editor. However, at the time there was minimal documentation available on how to use these modelling support tools and they were not fully mature, so that some archetype authoring needed to be carried out manually in the text file itself. The immaturity of the range of modelling tools available at the time was also reported by Buck et al. and Garde et al. [57, 63]. Good tool support is imperative to promote participation of non-technical domain experts in the design of high-quality archetypes.

However, the available range of tools has since increased and existing tools have matured. New versions of the *openEHR* Archetype Editor have been released, fixing some of the previous problems. In addition to the above editors, the LinkEHR Archetype Editor (LinkEHR-Ed)⁴ [73] has been made openly available. The LinkEHR-Ed Archetype Editor is an Open source multi-model archetype editor that currently works with standards CEN EN13606, CCR, CDA and proprietary XML-schemas like *openEHR*, epSOS, openMRS. Other archetype editors include that developed by the software firm Medical Objects⁵. Thus, some of the problems associated with the weaknesses and incompleteness of archetype editors have since and are still being addressed.

Lack of modelling guidelines or best practice rules

There are many different ways in which information can be modelled, and little guidance or rules for how to model information in archetypes are provided to help decide which ways are best. These issues concern common database modelling principles, for example how to model Boolean type information. While the *openEHR* specification provides for a Boolean data type (DV_BOOLEAN), which allows the values “true” and “false”, Boolean type information could just as well be modelled using structured text with the two values “true” or “false”. Similarly, there are many different ways to record a certain piece of information. For example, the presence of inflammation in a diagnosis archetype could be modelled by a DV_TEXT item called findings and the value to be filled into the archetype instance could be “Inflammation present”. Alternatively, the fact that inflammation is or is not present could be modelled explicitly with a DV_CODED_TEXT item called “Inflammation” with the values “Present” or “Not present” for choice at instantiation, or even with a DV_BOOLEAN item called “In-

⁴LinkEHR-Ed, http://www.e-recordsservices.eu/index.php?option=com_content&view=article&id=30&Itemid=52

⁵Medical Objects, <http://download.medical-objects.com.au/Template-Setup.exe>

flammation present” which could then either be “true” or “false”. There are currently no guidelines that help to decide which alternative should be used. The implications of the lack of modelling guidelines or rules will make it harder to maintain consistency and a high level of quality amongst archetypes, a weakness of the approach already identified by the NHS [243]. Modelling guidelines or modelling rules should be developed and made readily available. A recent conference paper by Kalra et al. addresses some of these issues [244].

Overlapping archetypes

In relation to the previous issue, archetypes that model the same or overlapping concepts between the two sources were found, for example the CLUSTER archetypes CLUSTER.body_site.v4 from the NHS repository versus CLUSTER.anatomical_location-precise.v1 from the *openEHR* repository, both of which are essentially modelling the same concept. Currently there is no governance process or authority that decides which of several possible archetypes describing basically the same concept should be used. In the absence of domain knowledge governance, the proliferation of incompatible, overlapping archetypes for related or the same concept will become an issue. In effect, using incompatible and overlapping archetypes only shifts the problem of semantic interoperability onto the archetype level, requiring the effort of establishing mappings between these archetypes. Thus, to avoid ‘rank growth’ of archetypes that could jeopardise the goal of achieving semantic interoperability promised by the *openEHR* archetype approach, there is a clear need for an internationally coordinated archetype development and maintenance process to provide quality assurance for archetypes [69, 68]. Garde et al. [69] suggest that ideally existing organisations should take on the task and responsibility of managing and publishing archetypes, such as national health information organisations for national scope archetypes and international organisations, such as the *openEHR* Clinical Review Board, or the Cochrane Collaboration⁶ for international level archetypes.

Furthermore, Buck et al. [57] present a model of distributed but cooperative archetype development and adoption to enable the international and inter-professional coordination of archetype development and maintenance. Their model is based on a process involving a hierarchical structure of professional committees, where every committee represents a particular division of the health domain and is responsible for a certain set of archetypes. However, while such a structured process may be desirable in the long run, some members of the archetype community feel that a degree of anarchy in archetype development may be a necessary phase in the beginning to test alternatives

⁶Cochrane Collaboration - International not-for-profit organisation preparing, maintaining and promoting the accessibility of systematic reviews of the effects of healthcare, www.cochrane.org/

in order to achieve high quality archetypes. The development of clinical archetypes is ongoing in many different domains, however, there is no real incentive to consolidate these efforts with isolated repositories of possibly incompatible and overlapping archetypes.

There are clearly risks associated with finalising archetypes too early, such as necessitating further changes to finalised archetypes that are not due to new knowledge creation, but to already existing knowledge. This may have a negative impact on downstream software, such as advanced information processing features, that rely on previous archetype versions. Adapting advanced information processing features to new versions can be a very resource consuming process, requiring, for example, re-design, re-implementation and re-testing of software. Thus, changes to archetypes should only be carried out by extension to cover the new knowledge and allow for backward compatibility. Input from researchers in clinical archetype design, especially regarding archetypes that overlap between clinical practice and research, e.g. pathology archetypes, may help in achieving a clinical archetype design that is also well suited for reuse in research. Similarly, it can be envisaged that research generated new data definitions, especially in the field of prognostic or diagnostic biomarkers may feed back into the clinical domain, possibly meaning that existing clinical archetypes may need to be augmented with these. Archetype governance approaches should take this into account. Overlaps are more likely to appear across repositories and due to this complicating issue, it was decided to mainly use the *openEHR* repository, which has been deemed sufficient for demonstrating the kind of overlap problems that may be encountered and how they may be solved. Thus, as a general comment, it is recommended that only archetypes from one source are used, such as the *openEHR* CKM, since the danger of overlap is more likely across different repositories than within the same repository and at least the archetypes will be consistent with one repository.

Small number of stable archetypes

The *openEHR* repository⁷ contains only a very small number of stable archetypes (only eleven by the time of writing), with the rest in team review state (15) or draft state (264). This means that over time archetypes used for modelling the information in the PCRC biobank archetype definitions changed, making the establishment of mappings difficult over time, since it was essentially chasing a moving target. Thus, for implementation purposes, it may be better to wait until a larger set of stable archetypes has become available. Archetypes in draft state may also contain inconsistencies, such as containing slots for archetypes that do not exist in the database.

⁷*openEHR* Knowledge Manager, <http://www.openehr.org/knowledge/>

Immaturity of the PCRC BIMS database schema

When this research was carried out, researchers of the biobank considered the modification of the schema and migration of existing data into a new database in the future, since the current schema did not adequately fulfil their information needs. The BIMS software used in the PCRC biobank also lead to a non-normalised design in the database, with similar information structures used across the database. Furthermore, it was found that different terms were used for the same concept and inconsistencies were found in handling the semantics of null values. Possibly a more mature database schema would have been easier to model with existing archetypes.

5.2.2 Difficulties identifying matching candidate archetypes

The process of identifying suitable candidate archetypes that contain elements that correspond to fields in the BIMS database and of establishing a mapping between these two presented considerable challenges. In most cases there was no obvious direct correspondence between the organisation of the data in the PCRC BIMS database and existing archetypes. Thus, the process of identifying archetypes that cover the fields in the PCRC BIMS database turned out to be tedious and time-consuming, since every data field needed to be searched for individually.

Currently, the only way provided to find out if specific elements are modelled in archetypes is by performing a string search online on the *openEHR* archetype repository, the Clinical Knowledge Manager (CKM)⁸ or on the downloaded NHS repository, or manually scanning through the archetype names in the hope that they will indicate possible containers of the desired element(s). Often only a small subset of elements covered by an archetype is needed, and it is not always clear from the name of the archetype whether or not it contains these elements. Furthermore, the archetype approach is still fairly new, not all clinical domains have yet been covered and only a relatively small number of stable archetypes exist.

This means that if a search does not return any results, it is difficult to know if no archetype yet exists that contains a particular element or if the containing archetype has not been found. There are many different ways of structuring and organising data and many alternative terms to describe the same concept or item, so it is easy to miss an archetype or an element in an archetype if it is described in an alternative way. As the number of archetypes increases, this problem will become more serious. This issue has also been identified by Buck et al. during the process of modelling an electronic patient record to support an innovative individual care concept for premature infants using the *openEHR* approach [57]. However, this problem may be aggravated in this

⁸CKM, <http://www.openEHR.org/knowledge/>

research due to the fact that clinical information about a patient in a biobank record for research is recorded and organised differently than the clinical information that is recorded for supporting patient-care. Existing archetypes have been created with a clinical, as opposed to a research setting in mind. Furthermore, the non-normalised design of the conceptual PCRC BIMS database schema meant that similar information structures were used across the database.

The approach proposed in this research, tackled this issue by including a step to first reorganise related information and arrange them into concepts, since archetype modelling is based on the notion of modelling information relating to one concept within the same archetype. In fact, the approach proposed during this research to model legacy information in the PCRC BIMS is essentially the same as the *openEHR* data modelling approach (*odma*) proposed by Buck et al. [57] to model the data in neonatology electronic patient record (EPR), which was published at the same time as this research was being carried out. Both approaches take into account that there is no “complete” set of archetypes to cover concepts in the clinical domain, but rather, only a small number exists and the majority still need to be developed. Thus, *odma* is a suitable approach to be used for modelling information in a legacy biobank database. In fact, the flow-chart shown in Figure 3.5 corresponds to steps two to four of the five-step *openEHR* data modelling approach. Furthermore, the information in the PCRC BIMS often contained more summary type information, or only snippets of information taken out of the context of the patient’s health record which are considered pertinent to the research being carried out, rather than the complete context. For example, one of the tables in the PCRC BIMS database, describing mainly the result of the pathological examination of a prostate biopsy, contained a field to describe if the result of digital rectal examination had been positive or negative, without any further context. This issue was partially alleviated by first reorganising the database fields into more archetype-friendly concepts, as described in the previous paragraph. Although clinical archetypes model generic concepts and thus should also be reusable even in a different context, they are structured and organised to support patient-care, rather than research.

Not being able to easily identify possible archetype candidates means that there is a danger that new archetypes are being developed for concepts that have already been modelled with archetypes, thus increasing the danger of creating overlapping and incompatible archetypes. This issue needs to be tackled by the archetype community, possibly by providing automated searching and mapping to save modelling effort as proposed by Buck et al. [57]. Some research groups are working on easing this process, e.g. Fernandez-Breis et al. [245] present a Semantic Web System, using OWL ontologies, for managing clinical archetypes in a repository, allowing archetypes to be

semantically annotated and semantically searched, but these functionalities are not available in the current archetype repositories.

5.2.3 Using clinical archetypes to describe clinical information in a research context

During the mapping process it was found that the information recording needs with regard to clinical information about a patient in a research context differ from the information needs with regard to clinical information to support patient care. This was found to complicate the process to use existing archetypes and a number of incompatibilities have been encountered. These include specific data recording restrictions due to data protection issues, the lack of providing explicit context to some of the information recorded, the use of relative timing for summary information and the recording of derivative, rather than original information. A short discussion on these issues is provided in the next few paragraphs.

Data protection

Due to data protection requirements, the recording of the patient's date of birth in the PCRC BIMS is not allowed as this is regarded as personal information and it is not permitted to induce any identifiable personal information in the biobank database (this does not apply to the date of death since data protection ceases at the death of the individual). However, due to the longitudinal aspect of the research carried out by the PCRC, the researchers need to know the age of the patient at certain times (at time of TURP (Transurethral resection of the prostate) and at the time of radical prostatectomy). In the case of the PCRC biobank, biosamples are collected during surgery and it is common practice to carry out age-matching for patients and their samples for specific studies. Thus, the age of the patient on the date of surgery is recorded both in the TURP and the radical prostatectomy tables. In the archetype-based EHR, the patient's birth date is recorded as part of the patient's demographic data and thus age can be inferred from the timing context that is mandated to be recorded within the COMPOSITION and some of the ENTRY archetypes (OBSERVATION, ACTION). As a workaround, an archetype to model the age of a patient (EVALUATION.age), to be used within any COMPOSITION, has been developed to record information that needs to be linked to the age of the patient on a particular date.

Timing context

openEHR has been designed to support clinical care, so the modelling of *openEHR* COMPOSITIONs divides between persistent and event type information. Thus, the

information to be recorded is either recorded under a persistent type COMPOSITION or an event type COMPOSITION. An example for information to be recorded under a persistent type COMPOSITION would be the patient's allergies, whereas the event type COMPOSITIONs would be used to record patient encounters or results of laboratory tests. Information that is event based, such as a bone scan and the interpretation of that bone scan is modelled in a way that mandates the recording of the exact context in time. Thus, the time relationship between two events recorded in an *openEHR*-based record can be deduced by comparing the timing context.

In the database of the PCRC BIMS, the exact context in time is not always recorded. Specifically, in several cases, the time context of some events, such as a bone scan and associated results or the administration of chemotherapy is modelled implicitly and relative to the particular event of a radical prostatectomy operation. In particular, the information in the PCRC BIMS database recorded in the "Patient Outcome" table and the two sub-tables "Treatment Options" and "Outcome" are implicitly assumed to contain information relating to events after the radical prostatectomy operation had been carried out. Similarly, it was found that any medication or therapy that the patient has received before radical prostatectomy is recorded in the "Current Medication" field in the "Patient" table, whereas information regarding the administration of radio/hormone or chemotherapy that the patient receives after radical prostatectomy is recorded in the "Treatment Options" table. This presents a problem when existing archetypes are to be used to present this information, since these have been modelled to mandate the recording of the timing information.

As a workaround to this problem it is proposed that the relative timing information be modelled with SECTION headings and that either the system implementation is relaxed to allow for required instance values to be missing or to use a dummy date that can be used to indicate the relative timing to be expressed outside the realm of archetypes. Alternatively, an integration archetype may be used to represent this information. None of these solutions is satisfactory as neither allows for safe sharing and the second one is only useful internally.

Derivative information

Three fields in the "Outcome" table described derivative information from the original patient record:

- Time to Hormone Refractory disease (Months)
- Time to Biochemical Recurrence (Months)
- Time to Clinical Recurrence (Months)

The pieces of information contained in the field name and instance value are difficult to deal with in an archetype environment. The problem is that they all implicitly refer to an event that has happened at a certain point in time - in this case the radical prostatectomy operation of the patient, i.e. the value entered into the database table describes the number of months between the radical prostatectomy operation and one of the three diagnoses made. The number of months has been derived by the research nurses from existing health record information detailing the timing of radical prostatectomy and the timing of the diagnosed problem (such as Hormone Refractory Disease, Biochemical Recurrence and Clinical Recurrence). In archetypes, problems and diagnoses are entered as instance information to the EVALUTION.problem or EVALUTION.problem-diagnosis archetypes, which also contain items to model the timing of the problem/diagnosis in the form of a date, which may be approximated from the instance information given in the PCRC BIMS database fields for these diagnoses. Thus, in an archetype-based system, the time interval between the radical prostatectomy and the specific later problem/diagnosis in an archetype context needs to be inferred by comparing the timing information recorded for the radical prostatectomy operation and the timing information for the diagnosis of any of these problems. This means that in this case, the extraction of the particular information (time in months between two events) is more complex than when the derived information is recorded directly as in the legacy database, for example requiring “canned queries”.

Implicit information

It was found that a lot of context was not explicitly recorded in the PCRC BIMS database, but assumed or known locally by the database users. For example, since the biobank is only concerned with prostate cancer, all sample donors will be male, a fact that is not explicitly stated in the database, since it can be assumed from the context. Similarly, as discussed in Chapter 3, the table named “BPH” in the PCRC BIMS database describes the tissue analysis of prostate tissue that was collected through a procedure called TURP. The database users know from other database users that the tissue was collected through TURP, but this fact is also not recorded in the database. This information was gathered during enquiries with the nurses who enter data into the database. However, it is this implicit, assumed or locally known information that needs to be recorded alongside the explicitly provided information to provide the context so that the information in the database becomes meaningful outside the local context, for example if the eBMRR for the PCRC biobank is meant to be accessible outside the realm of this biobank. Similarly, to facilitate intelligent reasoning or data mining, this type of information needs to be recorded explicitly. In the PCRC biobank, this context information has now been documented by the nurses and is supplied in form of an

accompanying Standard Operating Procedure (SOP) document for new database users. However, with appropriate archetypes, this information can be encoded alongside the explicitly modelled information. During this research, some attempt was made to explicitly model some of the implicit context in archetypes where feasible, for example to record TURP in an archetype detailing a procedure carried out on the patient, and to record BPH in the patient's diagnosis. Basically, the aim in this exercise was limited to contextualise the fields in the PCRC biobank, but no extra information was provided, for example to describe how the collected blood was processed into plasma and serum. This information is provided externally in SOPs, but should ideally be modelled and recorded as part of the record.

Summary of using clinical archetypes in a research context

During the mapping of the BIMS database to archetypes a one-to-one correspondence between the data that are recorded in the database and the ELEMENTs of archetypes was sought. However, the issues discussed in Section 5.2.3 show that such a one-to-one mapping is not always possible due to differences in the information recording needs or other conditions with regard to recording clinical information about a patient in a research versus a clinical context. In fact, such a one-to-one mapping may not always be desirable in disease registries when reusing clinical records for research purposes. For example, in some cases a research repository might only need to record that a clinical event has occurred at some time in the past or whether a particular laboratory test result was normal or abnormal, whereas clinicians may want to document the specific test values in detail. This presents an important challenge regarding the reuse of clinical records for research and thus for the design of archetypes. For example, the design of archetypes for recording the specific values of a laboratory test should possibly mandate the inclusion of a field to specify whether the recorded results are normal or abnormal, to widen the reuse possibilities of data recorded with this archetype for research purposes.

5.2.4 Issues mapping legacy database to archetypes

While the majority of the concepts in the database of the PCRC biobank could be covered with existing archetypes, it was often problematic to establish a structural mapping between the particular fields of the PCRC BIMS schema and elements of existing archetypes. This would not present a problem in a green-field situation, where the biobank is to be set up from scratch, but it does present a problem if the information is already stored in a proprietary way in an existing database and the information is to be made accessible through archetypes and templates (either by migrating all the

information from the legacy system into the *openEHR* archetype system, or by creating an archetype layer above the legacy database which is mapped to the underlying legacy database). There are a lot of biobanks in existence now, whose information is stored in proprietary implementations. Thus, to get an idea of the feasibility of representing the data in the database of an existing proprietary biobank information management system with *openEHR* archetypes, the issues encountered in establishing a structural mapping of the information of a legacy database to archetypes are discussed below, followed by a detailed discussion of each of them:

- Granularity of documentation
- Deduced information
- Modelling mismatches
- Terminologies and vocabularies
- Unstructured text
- Mandatory items
- Relationships

Granularity of documentation.

Differences of granularity of documentation between the PCRC BIMS database and existing archetypes created problems in mapping database fields to archetype elements. For example the existence of PIN (prostatic intra-epithelial neoplasia, a pathological condition) and its grade in resected prostate tissue is modelled in the PCRC BIMS table “Radical Prostatectomy” with two fields: one field labelled “PIN” that can carry the values “Yes” or “No”, and the field “PIN grade” that can carry the values “Low” or “High”. By contrast the existing archetype modelling the microscopic findings of resected prostate tissue only contains a field under “Additional findings” for which the value “High grade Prostatic intraepithelial Neoplasia” is given as a possible selection only. Furthermore, while in the original database, the non-existence of margins at the posterior location can be explicitly expressed by choosing the value “No”, this cannot be done explicitly within the archetype, only by assumption, i.e. if no margin is mentioned on the apical side, then it may be assumed that there is no margin on the apical side. The question arises if the negative of this information needs to be explicitly expressible, i.e. if a “Normal statement” is needed.

Deduced information.

Some information can be deduced from other information in the record and may not have been modelled explicitly in an archetype, thus preventing the establishment of a direct mapping. For example, the field “Time to process” in the PCRC BIMS urine and blood sample collection tables is not currently a field modelled in existing archetypes for sample collection and as such cannot be mapped. However, this information can be computed from the “Time of processing/spinning” minus “Time of voiding/collection”, which are modelled in the archetypes. The question is whether there is a semantic need to create an extra field in the corresponding archetype to model this information, for example using mechanism for calculating this value, or if the calculation should be carried out outside the realm of the archetype layer or to let the users deduce or calculate it themselves.

Differences in meta-modelling.

Problems were also encountered due to mismatches between meta-modelling and field-value modelling between candidate archetypes and the original database. This concerns situations where information given in a field or table name in the legacy database is not modelled with the same field name in the archetype, but as value in an archetype field with a more generic field name. This situation often leads to one-to-many relationships: to model one field from the database, several fields, possibly over several archetypes, may be needed. For example, to model the field “DRE” in the PCRC BIMS table “Biopsy” with the possible values “Positive” or “Negative” describing if any abnormality has been found during the DRE (Digital Rectal Examination) procedure, several items in several archetypes are needed, e.g. the field “Normal Statements” in the CLUSTER.palpation.v1 archetype set to either “Normal” or “Abnormal” and the “Object field” in the CLUSTER.exam-generic.v1 set to “rectum”. However, CLUSTER.exam-generic.v1 itself is embedded in OBSERVATION.exam.v1. This seems quite a convoluted way to model the required data. Alternative ways of modelling may be possible if some of the archetypes were adapted. For example, OBSERVATION.exam.v1 could be extended to contain a “State” attribute that contains a field to describe the examination technique, thus avoiding the need to embed the CLUSTER.palpation.v1 archetype to describe the above data. Another alternative is to modify the CLUSTER.palpation.v1 to contain a field for the object of examination (rectum), so that it can be embedded directly in the OBSERVATION.exam.v1 archetype, avoiding the need for the use of the CLUSTER.exam-generic.v1 archetype.

Similarly, in the case of the BIMS database field “Margins: Posterior” with a choice of “Yes” or “No” as possible values, the meta-data combines two items of in-

formation, the name of the histo-pathological finding (margin visible) and the location of this histo-pathological finding in the prostate (posterior), with the possible field-values either confirming or denying the existence of the margin at the posterior location in the prostate. To model this information with existing archetypes, one needs to use the CLUSTER.anatomical.location-precise.v1 archetype with the value of the item “Relative location:Aspect” set to “Posterior”, which is embedded in the CLUSTER.tumour_resection_margins.v1 archetype in the “Margin location” slot, which itself is embedded in the CLUSTER.microscopy_prostate_carcinoma.v1 archetype through the slot mechanism at the “Resection margin detail” slot under the “Surgical resection margin” cluster that is contained under the “Surgical resection margins” cluster in the CLUSTER.microscopy_prostate_carcinoma.v1 archetype. It is easy to see that the task to correctly model a field from the original database in the correct context with archetypes can become very complex.

Similarly, the table named “BPH” in the PCRC BIMS database describes the tissue analysis of prostate tissue that was collected through a procedure called TURP (Transurethral resection of the prostate) and which has resulted in the diagnosis of benign prostatic hyperplasia (BPH). In this case, the table name “BPH” encodes a diagnosis, i.e. benign prostatic hyperplasia, which can be recorded in the diagnosis field in the EVALUATION.problem-diagnosis.v1 archetype.

Terminologies and vocabularies.

The terms used in the PCRC BIMS database differed in several cases from those used in the archetypes in the public repositories. Thus, it is of utmost importance to obtain a detailed meta-data description of the fields in the source database to be able to determine if two different terms essentially mean the same thing (synonym), for example between “Involvement of Inflammatory Cells” (in the original database) and “Inflammation” (in the CLUSTER.microscopy_prostate_carcinoma.v1 archetype) or to differentiate between things that have the same name but mean different things (heteronym/homonym). While archetypes themselves are terminology neutral and can be developed independently of external terminologies, they can be linked to multiple external terminologies [38, 63, 72]. The issues encountered above highlight the importance of using the option of binding archetypes and data in archetypes to common medical vocabularies, such as SNOMED-CT or LOINC to uniquely define their meaning. Many of the terms in the PCRC BIMS database could be easily modelled with pre- or post-coordinated terms from medical terminologies (cf. [246, 247, 248]). Thus, it would have helped if both the archetypes and the original database had provided detailed term-bindings to a recognised medical terminology, such as SNOMED-CT, to ensure equivalence.

Unstructured text.

The data in the PCRC BIMS database was often stored in a free-text, unstructured form, whereas the archetype approach inherently promotes a very structured way of documenting clinical data in the EHR, which can present a problem when legacy records are to be converted into an archetype-based record, a problem previously recognised by Bird et al. [85]. For example, the fields “Current Medication”, “Current Medical History”, “Past Medical/Surgical History”, “Current Symptoms” and “Family History” all allow a free text entry, whereas corresponding archetypes follow a much more structured approach. Furthermore, sometimes the information provided as unstructured text in one field of the PCRC BIMS database may include such diverse data that it could be modelled with one of several possible archetypes. For example, five potential archetypes were found to model the medical histories provided as unstructured text in the “Current Medical History” or “Past Medical/Surgical History” field in the “Patient” table, depending on the particular content:

- EVALUATION.problem-diagnosis.v1
- EVALUATION.problem.v1
- EVALUATION.problem-genetic.v1
- EVALUATION.injury.v1
- EVALUATION.problem-diagnosis-histological.v1

An additional archetype, ACTION.procedure.v1 could be used to model the past surgical history potentially provided in the “Past Medical/Surgical History” field in the “Patient” table, increasing the number of possible archetypes that model the information in this particular field to six. Since a direct mapping between the content of the fields of an existing database to elements in an archetype is often not possible, this may represent a problem when, as part of converting to an archetype-based record or as part of exposing the legacy database through an archetype layer, the archetype-based database is to be populated with data from the original database or when the original database is to be queried through an archetype layer. In the future, advanced natural language processing techniques, information extraction techniques and semantic technology [249, 250, 251, 252, 253, 254, 255], may be used to automatically extract information from the unstructured text fields in an existing database and populate the correct archetype fields with it. However, advances in these fields are still very much confined to the research environment [253] and outside the scope of this research.

As a workaround to achieve a one-to-one mapping for integrating free text, summary type legacy data, the EVALUATION.clinical.synopsis archetype, which allows for summary information to be stored as free text can be used. However, this archetype does not improve interoperability and query ability of the information. Ideally, when the legacy data is migrated into the new database, someone would manually go through the content in each summary PCRC BIMS database field and distribute the information in free text format into the appropriate fields in the appropriate structured archetype.

Mandatory items.

Similar to mandating the provision of a timing context for information for certain classes of archetypes, such as OBSERVATION and ACTION, further pieces of information, such as information composer and committer are mandated for the recording of information with archetypes. This information is not provided or recorded in the PCRC BIMS database itself, but may be extracted from other sources, e.g. from log files that log the users that access the database and the actions they perform. Otherwise these fields need to be labelled “unknown”, when it is impossible to provide this information retrospectively. This problem was previously experienced also by Bird et al. [85].

Relationships

The tables in the PCRC BIMS database are conceptually related hierarchically with the “Patient” table acting as the root table and this relationship is reflected in their screen representation, for example, the table “Plasma” is instantiated through the table “Sample Collection - Blood”, reflecting the parent-child specimen relationship between blood and plasma. Internally, the Distiller software uses primary and foreign keys in the underlying relational database tables to link these tables. The *openEHR* specification defines the use of identifiers within the *openEHR* based record, specifying how instances of Compositions referring to one patient are linked to the record of that patient through the use of internal identifiers. However, any relationships between data in instances of Compositions within one patient’s record container are modelled on the data level. Therefore, the hierarchical relationship between the record of a parent sample and the record of a child sample, for example between blood and plasma and serum has been modelled through including the CLUSTER.citation archetype in the ADMIN_ENTRY.specimen archetype, which can be used to link to the record of the parent sample, because in the case of the PCRC BIMS database there is no sample ID given for all samples. Similarly, the hierarchical relationship between an instance of the Biochemistry table and instances of the Pre-Op/Post-Op PSA tables is also modelled by

including a CLUSTER.citation archetype in the EVALUATION.psa_velocity archetype that is used in the “Biochemistry.oet” template.

5.2.5 Summary of mapping issues

The mapping issues discussed above could be attributed to reusing archetypes that have been developed for use in the clinical context in a different context, the research context, although some may have been aggravated by the particular characteristics of the PCRC BIMS database. In particular, clinical archetypes have been developed to record information in clinical practice, for example during a doctor-patient encounter. Thus, these archetypes reflect the information needs and recording practices in clinical practice, such as requiring certain mandatory pieces of information not necessarily required in the research context, specifically context in time. Biobanks also need to record patient background information, such as disease, treatment and patient outcome, but while the domain of information to be recorded is overlapping, the recording needs and requirements differ slightly, as exemplified by the issue described in section “Deduced Information”. However, many of the discrepancies in the needs and modelling approaches encountered in this research between research and clinical practice also apply equally well to contrasting areas of clinical practice. For example, the recording needs between generalists and specialists, or between professions may be overlapping but the focus and detail required may be different, thus representing a common problem faced during archetype development in general.

In fact, many of challenges faced above are commonly encountered during the integration of distributed and heterogeneous information systems that have been designed for different use contexts. For example, Park and Ram [256] discuss several types of semantic conflicts that can occur and need to be mediated during system integration efforts. According to Park and Ram [256], semantic conflicts can either occur on the data level, such as data-value conflicts, data representation conflicts or data-unit conflicts etc., or on the schema level, such as naming conflicts, entity-identifier conflicts and schema-isomorphism conflicts etc. Using this classification, the issue described in “Granularity of documentation”, could be categorised as a data precision conflict, which is a data level conflict, issues “Metadata-level versus data-level modelling”, and “Unstructured text”, could be classified as schematic discrepancies, which are schema-level conflicts, and the issues described in “Terminologies and vocabulary”, are naming conflicts, which also occur on the schema level. An added problem with archetypes is that they represent a fixed target schema to which information needs to be mapped.

It has previously been shown that it is a non-trivial issue to upgrade a legacy system to an archetype-based system. Issues relating to this have been identified as early as in 2003 by Bird et al. [85] who describe their practical experiences won from two field

trials of exporting data from legacy systems into an archetype-based system. Several issues were highlighted. For example, the source system may not always provide data that is mandatory in the archetypes, so that this information needs to be inferred, such as units for quantities, scientific properties, person recording data. Problems were also encountered when the data in the source system was stored in less structured form, such as free-form text fields and the archetype provided structured fields. Another issue that added to the complexity of fitting in the legacy data into an archetype was when different ways of grouping data in target and source formats were used. Additional problems concerned inappropriate or wrongly placed data in the source database. They conclude that the rich functionality enabled by using the archetype approach, such as decision support systems and querying, cannot be exploited if the source systems do not provide the data in the appropriate form. Many of the same issues were encountered in this research, with the added complication of attempting a mapping of information recorded and used in a non-clinical context.

Today, most current EHRs and BIMS databases do not follow any recording standard but on the contrary use bespoke databases that have been designed specifically for the situation at hand and are only relevant to a particular hospital, study or biobank, often in proprietary format, so that reuse of a database in other contexts is not possible. The *openEHR* archetype approach offers a possible solution to the sharing of information between biobanks or between EHRs, but since most BIMS databases and EHRs are bespoke implementations, the effort involved in achieving a mapping between the existing databases to existing archetypes, even when all the archetypes needed to cover the fields in the databases have already been developed and are stable, can be considerable. However, if archetypes were only to be used in a “green-field situation”, i.e. in situations where new biobank databases or EHRs are being set up, then the wealth of information that already exists in databases could potentially be lost. In addition, the same conflicts may still arise in this situation, due to differences in information and documentation needs between healthcare facilities and biobanks, reconfirming the need for researchers to take part in the design of clinical archetypes to make them usable in the research context.

Research is ongoing on what approach would be best to upgrade legacy systems to the archetype approach. Chen et al. [84] report on their approach to automatically convert the content model of a regional EHR into *openEHR* archetype format and vice versa, preserving most of the structure and semantics of the original models. However, while this approach allows the underlying format to be expressed in archetype format, the resulting archetypes directly correspond to the data model of the clinical source systems (thus representing “integration archetypes”, which do not model data according to good archetype modelling principles, but rather only model the data in

the same format). This pushes the data interoperability issues to the archetype level, rather than solving them [85]. Thus, the problem of integrating legacy information with archetypes still needs to be solved.

Decisions need to be taken on a case-by-case basis if the effort required to properly transform legacy information into archetyped information is worth the effort, such as deconstructing unstructured text to fill appropriate fields in the archetypes rather than recording the unstructured text in a free text field, which limits the reusability of this information in a computational environment. On the positive side, the mapping effort only needs to be achieved once between each database and archetypes.

5.2.6 Further observations regarding new archetypes created in this research

Finally, the following observations have been made during this research. It was found that there was a much larger need for organisational level archetypes, rather than ENTRY and lower level archetypes, which specify the particular leaf data. Most of the new archetypes developed were organisational level or header type archetypes, such as COMPOSITION and SECTION archetypes, which correspond to document headings, and are generally used to organise the clinical information contained in ENTRY archetypes [44], making up 2/3 of the newly developed archetypes, with the other ones on the level of the ENTRY class in the archetype hierarchy. Also, the changes carried out on existing archetypes can be considered fairly minor except for the CLUSTER.microscopy_prostate_carcinoma.v1 archetype, which describes microscopic analysis of prostate cancer tissue. The legacy database provided much detailed information on microscopic analysis of prostate cancer tissue and most changes were carried out on the aforementioned archetype (see Table 3.2). This confirms that archetypes are generic and can be reused in to describe information in a non-clinical context.

5.3 Summary of modelling the information in the PCRC database with *openEHR* archetypes and templates

This part of the research provides an evaluation of using the *openEHR* archetype approach to model the information in an existing biobank. The questions to be answered were to what extent could existing archetypes be reused to model the information in the PCRC biobank and to identify the impact of modelling conflicts encountered on the integration of legacy databases. The aim was to try and model the information in the

database of an existing prostate cancer biobank, so that the information in this biobank can be completely represented with archetypes. This represents the first attempt to apply the archetype approach to model the information in a biobank database and to reuse archetypes that have been developed for the creation of EHRs in the context of a biobank. A major advantage offered by the use of EHRs is that they support the reuse of vast amounts of clinical information including outcomes data for different purposes, such as research. The results of this research provide positive evidence to support this potential in the context of biomedical research.

In this regard it was very promising that the majority of the fields in the PCRC BIMS database, including sample-related fields, could be represented with existing archetypes which could be reused without change, thus reducing modelling effort. This indicates that the underlying principle of reuse of archetypes is valid and that archetypes developed to represent clinical concepts in EHRs can generally be reused to represent clinical concepts in biobanks. Many archetypes originally developed for the Electronic Health Record (EHR) domain can be re-used to model the clinical/phenotypic and sample information in the biobank context, which validates the genericity of these archetypes and their potential for re-use in the context of biomedical research. Some archetypes needed to be extended and modified, which shows that archetypes are still not mature and a set of high-quality archetypes with which to model clinical data is yet to be achieved. Real-world case studies such as this one can help improve clinical archetypes to make them reusable in different contexts, such as within the research context in the eBMRR, thus promoting the reuse of clinical information in research. It is expected, with the increased interest in the archetype approach, that both the quality of available clinical archetypes and that the number of concepts modelled with archetypes will increase. If the *openEHR* approach becomes a widely used approach for EHRs, then it may even become possible to import background information from a patient directly from the EHR of the patient. Basing the proposed eBMRR on the same technology as is currently being proposed for EHRs provides a parallel solution for sharing information amongst collaborating biobanks, but it also allows for the possibility of importing relevant background or phenotypic information about the patient directly from an archetype-based patient's/donor's EHR and vice versa.

As may be expected with a novel approach like this, some issues were encountered during this research. For example, problems were encountered when using the modelling support tools. Similarly, a lack of modelling guidelines or best practice rules made it difficult to design high-quality archetypes. Both these issues are currently being addressed. For example, modelling support tools have matured since the start of this research in 2008, providing a bigger choice and more functionality, together with online tutorials on how to use them, although further improvements can be made. A

recent paper by Kalra et al. provides some modelling guidelines [244].

Furthermore, the problem of overlapping archetypes was encountered, resulting from the lack of domain knowledge governance. However, the majority of problems were encountered during the mapping of the PCRC BIMS database to archetypes. First of all, the process of identifying archetypes that contained the desired items and the mapping of the fields of the PCRC BIMS database to items in existing archetypes proved to be quite challenging. This was partially due to the fact that the process of finding archetypes that contained certain items is still quite manual. This problem was also aggravated by the fact that the archetype approach was used on a database of a biobank information management system in which information is organised differently from how it is organised in a conventional EHR, often focussing on summary information or snippets pertinent to the research carried out, thus making it harder to identify suitable archetypes.

While an archetype-based approach to represent the clinical information in a biobank is suitable and development effort for new biobank information management systems can be reduced by reusing generic archetypes, some issues were encountered during the process of mapping the legacy database fields to leaf data types in existing archetypes. These issues include differences in the granularity of documentation, the issue of how to deal with information that can be deduced from the context, mismatches between schema and data modelling, differences in terminologies and vocabularies used, differences in structural organisation and differences in mandatory information, many of which represent common system integration challenges. These issues mean that a direct one-to-one mapping could not be established between fields in the underlying database and fields in archetypes. The consequence of these issues is that the live integration of information from the legacy database into an archetype-based context may not be possible and that migration of legacy data into an archetype-based system using universally valid archetypes may not be 100% achievable. Thus, in addition to universally valid archetypes, some ‘only locally valid’ integration archetypes may need to be used to be able to make this information available in an archetype-based system, so that information is not misrepresented or represented in the wrong context. Thus, in this case the recommendation is to migrate the legacy data into an archetype-based system, using a semiautomatic process for those fields for which a direct mapping could be established and using human intervention to deal accordingly with those fields that could not be mapped in a direct one-to-one manner.

The value of the archetype approach as a candidate for information sharing between hospitals or biobanks could be much increased if the conversion process between existing data sources to universally accepted archetypes, once they become available, could be eased, e.g. through appropriate annotations using common medical terminologies in

both the existing system and archetypes and the development of automated mapping tools. Possibly, research into solving semantic interoperability conflicts to facilitate the integration of distributed and heterogeneous information systems [256] could be leveraged for this.

Furthermore, there is a need for the development and management of high-quality archetypes to cover more clinical and research domain knowledge. However, even with a set of high-quality archetypes, adding archetype-support to an existing system will remain a challenge, since differences in information modelling between the existing data source (which may also be flawed or may not follow good database design principles) and the archetypes will often prevent a straightforward one-to-one mapping of fields between the two, even though the archetype is representing the correct concept.

The process of archotyping also showed flaws in the legacy database, such as using different terms for the same concept and inconsistencies in handling the semantics of null values. The archetype approach is designed to help avoid such issues and applying the archetype approach to the database of the PCRC biobank provided a common basis for discussion about what kind of information should be recorded and how, independent of technical implementation details.

The outcome of this evaluation is thus that the *openEHR* approach presents a suitable approach for the development of the eBMRR and it is very promising that existing archetypes can be reused to cover the clinical information needed in a biobank, although integration of legacy data may be problematic. However, in a green-field approach, by reusing existing archetypes, modelling effort can be significantly reduced and information is thus automatically recorded in an interoperable manner that allows for its safe sharing. With the ongoing development of archetypes their quality is expected to improve and more of the clinical information will be covered, so that in future all of the clinical information may be covered by using existing archetypes.

5.4 Evaluation of modelling omic information with the *openEHR* archetype methodology

This section provides an evaluation of modelling omic information with the *openEHR* archetype modelling approach using standard omic data formats as the basis for the development of omic archetypes. The development of omic archetypes that model the execution and results of several omic experiments has been described in chapter 4. Specifically, omic archetypes were successfully developed to model mass spectrometry experiments, based on mzML, general functional genomic (i.e. omic) experiments, based on the FuGE data format and gel electrophoresis experiments, based on GelML,

an extension of FuGE.

In this section, the syntactic and semantic validity of the omic archetypes developed are discussed. The archetypes are evaluated with regard to whether they can be used to faithfully record the same information as the source omic data formats. Specifically, the preservation of the original context and meaning of a data value as recorded in an instance of the original data format is evaluated, upon recording the same data value with the omic archetypes that have been developed based on the original data format. Furthermore, the usefulness of 2-phase approach that was developed to facilitate the development of omic archetypes from standard omic data formats is discussed, in particular with regard to the feasibility of applying this approach for the development of further omic archetypes from other omic data formats. Finally, a discussion on the limitations of applying the *openEHR* archetype approach, which was developed specifically to model clinical information in archetypes to support the development of interoperable EHRs, to model omic research information is provided. In relation to this, a discussion on the limitations of using a patient-centric approach in an investigation-based environment is also provided.

5.4.1 Syntactic validity of the developed omic archetypes

Syntactic correctness in this case refers to using the correct ADL syntax in the archetypes developed. This for example includes that all mandatory sections are present (for example the description and ontology sections), that the filename and archetypeID correspond, that all opening brackets are closed, that all node identifiers are unique and that each node is accompanied by a description in the ontology section in the ADL file.

Generally, archetypes are developed with an archetype editor, which provides an inbuilt archetype editor and thus automatically ensures that the serialised ADL files of archetypes developed with the editor are syntactically correct. However, during this research, candidate archetypes are created using bespoke transformation software that is designed to create archetypes as in-memory objects, using the Java reference implementation of the *openEHR* reference model. To serialise the archetype objects to ADL files, the ADL serializer provided as part of the Java reference implementation is used, so syntactic correctness for the candidate archetypes is automatically ensured (assuming that the ADL serializer works correctly).

However, substantial modifications, are carried out on the candidate archetypes during the semantic review process. Currently available archetype editors do not support working on several archetypes concurrently and lack a drag-and-drop functionality to allow the copying of hierarchy fragments and all associated parts from one archetype into another. Therefore, these semantic modifications were carried out using a simple

text editor. The general immaturity of archetype modelling tools has already been discussed in Section 5.2.1. In addition, it would also be useful for future versions of archetype editors to allow for editing across several archetypes at the same time and to support a drag-and drop functionality between archetypes, for example when (fragments of) the contents of two or more archetypes are to be merged. This means that syntax errors, such as duplicating node identifiers or forgetting to add a description to a newly added node, may have been introduced. Syntactic validation can easily be carried out computationally and in fact the *openEHR* archetype editor carries out a syntactic check upon trying to open an archetype file. The archetype editor only opens syntactically correct archetypes and provides clues as to the syntactic problem encountered when it tries to open a syntactically incorrect archetype. Thus, all of the modified archetypes were validated syntactically by opening them in the *openEHR* archetype editor and correcting any errors that were detected.

During this process, small incompatibilities between the ADL serializer provided by *openEHR* and the archetype editor provided by *openEHR* were discovered. For example, some constructs created automatically by the ADL serializer were not validated by the *openEHR* archetype editor, for example, empty terminology descriptions (in the ontology section of the ADL archetype: `terminologies_available = <...>`) were not accepted, so this line was deleted in the offending archetypes for which no terminology mappings are provided. Similarly, problems were encountered when an `ELEMENT` restricted to a `DV_PROPORTION` value did not specify a denominator. These and similar issues were corrected manually.

Later, the *openEHR* ADL workbench⁹ was used for validation support (it also provides an inbuilt archetype validator). However, further differences were encountered between the validation of archetypes that appear valid in the *openEHR* Archetype Editor and the *openEHR* ADL workbench. In particular, the ADL workbench validator was more strict, and many of the *openEHR* provided archetypes do not pass validation with this validator due to the validator being stricter with regard to the syntactic validation criteria. Since it is not the focus of this research to correct the syntax of existing archetypes, it was only used to validate that certain structures not supported in the *openEHR* Archetype Editor were correct. In particular, the *openEHR* Archetype Editor did not recognise “`use_node`” constructs to `CLUSTER` nodes and also did not support the “`summary`” attribute of the `HISTORY` class. Thus, two versions of archetypes are provided where these conflicts arose: one that can be opened in the *openEHR* Archetype Editor, containing dummy object constraints for those constructs that the editor does not support and one containing the correct constructs, which were

⁹ADL workbench, http://www.openehr.org/svn/ref_impl_eiffel/TRUNK/apps/adl_workbench/doc/web/index.html

validated separately in the ADL workbench. This is another example of the immaturity of the *openEHR* support tools and needs to be addressed.

5.4.2 Semantic validity of the developed omic archetypes

With regard to semantic validation, the criteria provided in the “Editorial checklist for clinical content review of archetypes”¹⁰ provided by *openEHR* were applied during phase 2 of the omic archetype development process.

One of the main criteria for archetypes is that an archetype shall describe a single (clinical) concept without overlaps to other archetypes, based on consensus, with focus on universal use case [44, 38]. Adapted to omic content, an archetype should describe a single “omic” concept without overlaps to other archetypes, based on consensus. Other criteria listed in this checklist include checking the appropriateness and completeness of data elements and attributes, naming, descriptions, slots and meta-data as well as correctness of cardinality and occurrence constraints in each particular case.

Omic data formats have, like clinical archetypes, been developed by groups of experts from different professional backgrounds and cultures with the aim of supporting the safe sharing of omic, or clinical information, respectively. However, as discussed in Chapter 4, omic data formats to support the sharing of omic information are designed according to different principles than *openEHR* archetypes. As discussed in Chapter 4, omic data standards are minimum information standards, but the corresponding data formats generally allow for more (or less) information to be recorded. It is assumed that the original consensus-based data formats model sufficient information needed to support the adequate recording of a particular omic experiment. It is outside the scope of this thesis to evaluate the quality of the source data formats and whether they are capable of adequately describing omic research data.

During the omic archetype development process, the faithfulness of the automatic transformation software was tested to ensure that the mappings were applied correctly during the transformation process. Visual inspections of candidate archetypes were carried out to ensure that their content corresponded exactly to the content of the original data formats, as dictated by the mapping, such as providing the correct occurrence constraints, naming, data types and the inclusion of mapped ontology elements. Thus, by assuming that the information in the original data formats is sufficient and correct and the mappings created are sensible, then one can assume that corresponding constraints on objects and attributes in the candidate archetypes, including naming, descriptions, slots and meta-data as well as cardinality and occurrence constraints, as created by the transformation software, are correct.

¹⁰Editorial checklist, <http://www.openEHR.org/wiki/display/healthmod/Editorial+checklist+for+clinical+content+review+of+archetypes>

However, archetype design is controlled by semantic requirements and principles, such as using correct *openEHR* datatypes, adherence to the single-concept paradigm, avoiding overlaps and modelling information on the ontologically correct archetype class level. Therefore, modifications were carried out on the candidate archetypes during phase 2 of the archetype development process to satisfy these principles, involving the copying and pasting of parts of ADL files. The original data formats dictate both the information that needs to be modelled and the relationship between different data values. The data formats thus provide both the basis for developing archetypes as well as for the templates where the archetypes are recomposed to model the hierarchical relationships between data. While care was taken to preserve the original hierarchical relationships as used in the XML schema where possible and meaningful, the design of the source XML schema are not guided by the same principles as archetype design. As a consequence, substantial structural modifications needed to be carried out on many of the automatically created candidate archetypes to produce a set of reusable archetypes that fulfil semantic archetype design principles. These structural modifications specifically concern the changes around containment hierarchies and composite associations of FuGE and FuGE-based candidate archetypes and may have changed original hierarchies to such an extent as to cause problems with regard preserving the original meaning and context of data as intended by the original data formats.

In the absence of well-defined archetype development guidelines and the vagueness and ambiguity of the boundaries of one concept, specifically around concepts and sub-concepts, some may not agree with the resulting archetypes as they may have a different perspective on the boundaries of a specific concept. Since the meaning of what constitutes a “concept” can be ambiguous and thus interpreted differently, it provides little guidance in the development of archetypes. Often, existing archetypes were consulted to find solutions for the detailed modelling of information within archetypes. An equally valid set of differently organised archetypes could have been developed to model the knowledge contained in the original data formats. The omic archetypes developed here thus represent a best attempt at reconciling the knowledge contained in original data formats with *openEHR* archetype design principles. Furthermore, the omic archetypes developed in this research are based on only a small number of data formats, and thus may not model maximum data sets for the particular concept that they model, since they originated from domain-specific data formats. As part of future research, the archetypes created in this research should be reviewed by a representative group of biomedical research scientists to determine whether their current structural organisation is sensible and whether their content can be considered complete in a global context.

Thus, while there are many different and equally valid ways in which information

may be organised within the realm of archetypes, the archetypes developed in this research need to be evaluated as to whether they faithfully model the knowledge contained in the original data formats. As discussed above, extensive structural reorganisation of the automatically created candidate archetypes were carried out, changing some of the original hierarchical relationships. Therefore, the archetypes need to be tested with regard to their ability to be used to record the same information as in the original data format and that the context and meaning of this information is preserved as intended by the original data format. To evaluate the omic archetypes created in during this research as described in the previous chapter, instance files for the original data formats are obtained and the omic archetypes are arranged into templates, using the Ocean Informatics Template Designer¹¹, that can be further customised to the data recorded in these instance files. Examples of mappings, using Xpath, from the data recorded in the original data format to the corresponding leaf nodes in the templates, using the customised omic archetypes, are provided. These mappings demonstrate the faithfulness of the developed archetypes to record the same information as in the original data formats and show that the original context is preserved. This evaluation provides a validation of the faithfulness of the 2-phase process for developing omic archetypes from omic data formats. It gives an insight into whether the 2-phase omic archetype development process can be used to create archetypes that faithfully model the knowledge contained in the original data formats, and whether modifications carried out on the candidate archetypes compromise the completeness and correctness of the underlying data standards, with regard to preserving the context and meaning of the data as recorded with the original data format. The evaluation is carried out in the context of the specific data formats used as basis for the development of these archetypes in this research, specifically with regard to mzML, the standard data format for mass spectrometry, and GelML, the standard data format for electrophoresis (GelML), validating the corresponding mass spectrometry archetypes and the gel electrophoresis archetypes, respectively. An abbreviated evaluation is provided for FuGE-based archetypes, since no instance file for FuGE could be obtained.

5.4.3 Evaluation of mass spectrometry archetypes

To validate the mass spectrometry archetypes developed from the mzML data format, they have been arranged into a compositional hierarchy in a template, as dictated by the slot constraints of the individual archetypes, which largely follows the same hierarchy as in the original XML schema, but without defining any further constraints. The result is a template that is constrained in the hierarchy of the particular archetypes

¹¹Ocean Informatics Template Designer, www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-template-designer.html

used within it, but otherwise minimally constrained and thus can be used to record the execution and result of any mass spectrometry experiment. This template file is provided on the attached CD.

A number of mzML instance files can be found on the PSI website¹². In addition, Proteome Commons¹³ provides a number of mzML test instance files.

Two instance files were selected, “neutral_loss_example_1.1.0.mzML” and “MRM_example_1.1.0.mzML” from the HUPO PSI website¹⁴. These instance files are also provided on the attached CD.

The following listings demonstrate the mapping of specific data recorded in the mzML instance files to the leaf nodes of the archetypes used in this template, using XPath.

Listing 5.1 shows an extract from the “neutral_loss_example_1.1.0.mzML” instance file and Listing 5.2 maps some of these instances to leaf nodes in the “Mass Spectrometry Experiment.oet” template.

```

1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!-- Hand crafted example file with a neutral loss spectrum -->
3 <mzML xmlns="http://psi.hupo.org/ms/mzml" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
4   xs:schemaLocation="http://psi.hupo.org/ms/mzml http://psidev.info/files/ms/mzML/xsd/mzML1.1.0.
5   xsd" accession="plgs.example" version="1.1.0" id="urn:lsid:proteios.org:mzml.plgs.example">
6   <cvList count="2">
7     <cv id="MS" fullName="Proteomics Standards Initiative Mass Spectrometry Ontology" version="1.18.1"
8       URI="http://psidev.cvs.sourceforge.net/*checkout*/psidev/psi/psi-ms/mzML/
9       controlledVocabulary/psi-ms.obo"/>
10    <cv id="UO" fullName="Unit Ontology" version="1.20" URI="http://obo.cvs.sourceforge.net/*checkout
11      */obo/obo/ontology/phenotype/unit.obo"/>
12  </cvList>
13  <fileDescription>
14    <fileContent>
15      <cvParam cvRef="MS" accession="MS:1000326" name="constant neutral loss spectrum"/>
16    </fileContent>
17    <sourceFileList count="2">
18      <sourceFile id="SF1" name="test.RAW" location="file:///F:/data/testdir">
19        <cvParam cvRef="MS" accession="MS:1000563" name="Thermo RAW file" value="" />
20        <cvParam cvRef="MS" accession="MS:1000768" name="Thermo nativeID format"/>
21        <cvParam cvRef="MS" accession="MS:1000568" name="MD5" value="7a09cc8a55aaca14741ca0777658e496"/>
22      </sourceFile>
23    </sourceFileList>
24  </fileDescription>
25  <softwareList count="1">
26    <software id="Xcalibur" version="2.0.5">
27      <cvParam cvRef="MS" accession="MS:1000532" name="Xcalibur" value="" />
28    </software>
29  </softwareList>
30  <instrumentConfigurationList count="1">
31    <instrumentConfiguration id="TSQ">
32      <cvParam cvRef="MS" accession="MS:1000199" name="TSQ Quantum" value="" />
33      <cvParam cvRef="MS" accession="MS:1000529" name="instrument serial number" value="23433"/>
34    <componentList count="5">
35      <source order="1">
36        <cvParam cvRef="MS" accession="MS:1000398" name="nanoelectrospray" />
37      </source>
38      <analyzer order="2">
39        <cvParam cvRef="MS" accession="MS:1000081" name="quadrupole" value="" />
40      </analyzer>
41      <analyzer order="3">
42        <cvParam cvRef="MS" accession="MS:1000081" name="quadrupole" value="" />
43      </analyzer>
44      <analyzer order="4">

```

¹²mzML, <http://www.psidev.info/mzml>

¹³10 real sample spectra files, available on <https://proteomecommons.org/dataset.jsp?i=73062>

¹⁴mzML, <http://www.psidev.info/mzml>

```

40     <cvParam cvRef="MS" accession="MS:1000081" name="quadrupole" value=""/>
41 </analyzer>
42 <detector order="5">
43     <cvParam cvRef="MS" accession="MS:1000253" name="electron multiplier" value=""/>
44 </detector>
45 </componentList>
46 <softwareRef ref="Xcalibur"/>
47 </instrumentConfiguration>
48 </instrumentConfigurationList>
49 <dataProcessingList count="1">
50     <dataProcessing id="XcaliburProcessing">
51         <processingMethod order="1" softwareRef="Xcalibur">
52             <cvParam cvRef="MS" accession="MS:1000035" name="peak picking"/>
53         </processingMethod>
54         <processingMethod softwareRef="Xcalibur" order="2">
55             <cvParam cvRef="MS" accession="MS:1000544" name="Conversion to mzML"/>
56         </processingMethod>
57     </dataProcessing>
58 </dataProcessingList>
59 <run id="Exp01" defaultInstrumentConfigurationRef="TSQ" defaultSourceFileRef="SF1" startTimeStamp="
    2008-06-27T15:23:45.00035">
60     <spectrumList defaultDataProcessingRef="XcaliburProcessing" count="1">
61         <spectrum index="0" id="controller=0 scan=1" defaultArrayLength="18">
62             <cvParam cvRef="MS" accession="MS:1000326" name="constant neutral loss spectrum"/>
63             <cvParam cvRef="MS" accession="MS:1000127" name="centroid spectrum"/>
64             <cvParam cvRef="MS" accession="MS:1000130" name="positive scan" value=""/>
65         <scanList count="1">
66             <cvParam cvRef="MS" accession="MS:1000795" name="no combination" value=""/>
67         <scan instrumentConfigurationRef="TSQ">
68             <!-- the value of the neutral loss -->
69             <cvParam cvRef="MS" accession="MS:1000803" name="analyzer scan offset" value="80" unitCvRef="
                MS" unitAccession="MS:1000040" unitName="m/z"/>
70         <scanWindowList count="1">
71             <scanWindow>
72                 <cvParam cvRef="MS" accession="MS:1000501" name="scan window lower limit" value="400"
                    unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
73                 <cvParam cvRef="MS" accession="MS:1000500" name="scan window upper limit" value="1600"
                    unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
74             </scanWindow>
75         </scanWindowList>
76     </scan>
77 </scanList>
78 <precursorList count="1">
79     <precursor>
80         <isolationWindow>
81             <!-- Q1 transmission window -->
82             <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset" value="1.0"
                unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
83             <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset" value="1.0"
                unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
84         </isolationWindow>
85         <activation>
86             <cvParam cvRef="MS" accession="MS:1000045" name="collision energy" value="35" unitCvRef="UO"
                unitAccession="UO:0000266" unitName="electronvolt"/>
87             <cvParam cvRef="MS" accession="MS:1000133" name="collision-induced dissociation" value=""/>
88         </activation>
89     </precursor>
90 </precursorList>
91 <productList count="1">
92     <product>
93         <isolationWindow>
94             <!-- Q3 transmission window -->
95             <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset" value="1.0"
                unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
96             <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset" value="1.0"
                unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
97         </isolationWindow>
98     </product>
99 </productList>
100 <binaryDataArrayList count="2">
101     <binaryDataArray encodedLength="96">
102         <cvParam cvRef="MS" accession="MS:1000514" name="m/z array" unitCvRef="MS" unitAccession="
            MS:1000040" unitName="m/z"/>
103         <cvParam cvRef="MS" accession="MS:1000576" name="no compression"/>
104         <cvParam cvRef="MS" accession="MS:1000521" name="32-bit float"/>
105         <binary>fZ/QQ0wn2UMpzOFDqBbpQw5VD...</binary>
106     </binaryDataArray>
107     <binaryDataArray encodedLength="96">

```

```

108     <cvParam cvRef="MS" accession="MS:1000515" name="intensity array" unitCvRef="MS" unitAccession
109         ="MS:1000131" unitName="number of counts"/>
110     <cvParam cvRef="MS" accession="MS:1000576" name="no compression"/>
111     <cvParam cvRef="MS" accession="MS:1000521" name="32-bit float"/>
112     <binary>AKDIRACAn0QAAlhEAEBYRACAy...</binary>
113 </binaryDataArray>
114 </binaryDataArrayList>
115 </spectrum>
116 </spectrumList>
117 </run>
</mzML>

```

Listing 5.1: neutral_loss_example_1.1.0.mzML

```

<!-- Lines 26 - 47 (instrumentConfiguration)-->
[openEHR-EHR-COMPOSITION.report.v1]
/content [openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/protocol[at0166]
/items [openEHR-EHR-CLUSTER.device-mass_spectrometer.v1 and name/value='Mass Spectrometer']
/Spectrometer'
/items[at0002] (Manufacturer model name) = "TSQ Quantum"
/items[at0006] (Serial number) = "23433"
/items [openEHR-EHR-CLUSTER.software-mass_spectrometry.v1 and name/value='Mass Spectrometry
Software']
/items[at0002] (Name) = "Xcalibur"
/items[at0003] (Version) = "2.0.5"
/items [openEHR-EHR-CLUSTER.device-mass_spectrometer-source.v1 and name/value='Mass
Spectrometer Source']
/items[at0.221] (Ionization type) = "nanoelectrospray"
/items[at0.321] (Order) = "1"
/items [openEHR-EHR-CLUSTER.device-mass_spectrometer-analyzer.v1 and name/value='Mass
Spectrometer Analyzer'] [1]
/items[at0003.1] (Mass analyzer type) = "quadrupole"
/items[at0.348] (Order) = "2"
/items [openEHR-EHR-CLUSTER.device-mass_spectrometer-analyzer.v1 and name/value='Mass
Spectrometer Analyzer'] [2]
/items[at0003.1] (Mass analyzer type) = "quadrupole"
/items[at0.348] (Order) = "3"
/items [openEHR-EHR-CLUSTER.device-mass_spectrometer-analyzer.v1 and name/value='Mass
Spectrometer Analyzer'] [3]
/items[at0003.1] (Mass analyzer type) = "quadrupole"
/items[at0.348] (Order) = "4"
/items [openEHR-EHR-CLUSTER.device-mass_spectrometer-detector.v1 and name/value='Mass
Spectrometer Detector']
/items[at0003.1] (Detector type) = "Electron multiplier"
/items[at0.382] (Order) = "5"
/items [openEHR-EHR-CLUSTER.device-details-mass_spectrometer.v1 and name/value='Mass
Spectrometer']

<!-- Line 60 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content [openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/protocol[at0166]
/items[at0177] (Default Data Processing)
/items[at0178] (Processing method)
/items[at0180 and name/value='Data processing action'] = "peak picking"
/items [openEHR-EHR-CLUSTER.software-mass_spectrometry.v1 and name/value='Mass Spectrometry
Software']
/items[at0002] (Software name) = "Xcalibur"

<!-- Lines 61 - 64 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content [openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/data[at0001]
/events[at0002]
/data[at0003]
/items [openEHR-EHR-CLUSTER.mass_spectrum.v1]
/items[at0001] (Spectrum type) = "constant neutral loss spectrum"
/items[at0019] (Spectrum representation) = "centroid spectrum"
/items[at0035] (Scan polarity) = "positive scan"

<!-- Lines 65 - 77 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content [openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/data[at0001]
/events[at0002]

```

```

/state[at0006]
/items[at0007] (Spectrum settings)
/items[at0123] (List of Scans)
/items[at0124] (Spectra combination) = "no combination"
/items[at0129] (Scan)
/items[at0138] (Analyzer scan offset) = "80 m/z"
/items[at0152] (Scan window list)
/items[at0153] (Scan window)
/items[at0155] (scan window lower limit) = "400 m/z"
/items[at0154] (scan window upper limit) = "1600 m/z"
etc.

```

Listing 5.2: Examples of XPath to leaf nodes to instance data stored in neutral_loss_example_1.1.0.mzML

A screenshot of the template with line number corresponding to Listing 5.1 is shown in Figure 5.1.

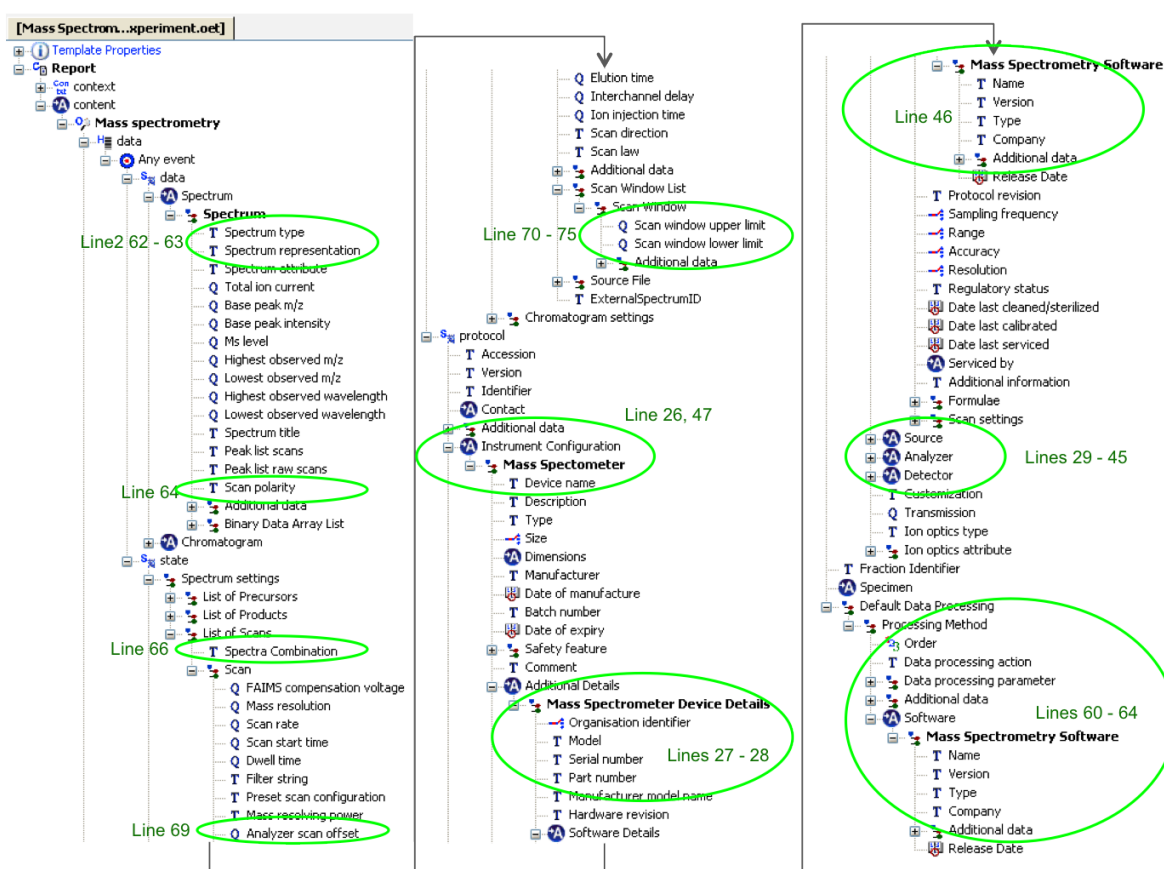


Figure 5.1: Screenshot of a template for a spectrum example; line numbers refer to Listing 5.1.

Listing 5.3 shows an extract from the “MRM_example_1.1.0.mzML” instance file and Listing 5.4 maps some of these instances to leaf nodes in the “Mass Spectrometry Experiment.oet” template.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Hand crafted example of SRM encoding in mzML. Contains two transitions -->
3 <mzML xmlns="http://psi.hupo.org/ms/mzml" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
  xs:schemaLocation="http://psi.hupo.org/ms/mzml http://psidev.info/files/ms/mzML/xsd/mzML1.1.0.
  xsd" accession="mrm_example" version="1.1.0" id="urn:lsid:proteios.org:mzml.plgs.example">

```



```

4 <cvList count="2">
5   <cv id="MS" fullName="Proteomics Standards Initiative Mass Spectrometry Ontology" version="2.17.1"
      URI="http://psidev.cvs.sourceforge.net/*checkout*/psidev/psi/psi-ms/mzML/
      controlledVocabulary/psi-ms.obo"/>
6   <cv id="UO" fullName="Unit Ontology" version="1.20" URI="http://obo.cvs.sourceforge.net/*checkout
      */obo/obo/ontology/phenotype/unit.obo"/>
7 </cvList>
8 <fileDescription>
9   <fileContent>
10    <cvParam cvRef="MS" accession="MS:1001473" name="selected reaction monitoring chromatogram"/>
11  </fileContent>
12  <sourceFileList count="2">
13    <sourceFile id="SF1" name="test.RAW" location="file:///F:/data/testdir">
14      <cvParam cvRef="MS" accession="MS:1000563" name="Thermo RAW file" value=""/>
15      <cvParam cvRef="MS" accession="MS:1000568" name="MD5" value="7a09cc8a55aaca14741ca0777658e496"/>
16      <cvParam cvRef="MS" accession="MS:1000768" name="Thermo nativeID format"/>
17    </sourceFile>
18  </sourceFileList>
19 </fileDescription>
20 <softwareList count="1">
21   <software id="Xcalibur" version="2.0.5">
22     <cvParam cvRef="MS" accession="MS:1000532" name="Xcalibur" value=""/>
23   </software>
24 </softwareList>
25 <instrumentConfigurationList count="1">
26   <instrumentConfiguration id="TSQ">
27     <...>
28   </instrumentConfiguration>
29 </instrumentConfigurationList>
30 <dataProcessingList count="1">
31   <dataProcessing id="XcaliburProcessing">
32     <processingMethod order="1" softwareRef="Xcalibur">
33       <cvParam cvRef="MS" accession="MS:1000035" name="peak picking"/>
34     </processingMethod>
35     <processingMethod softwareRef="Xcalibur" order="2">
36       <cvParam cvRef="MS" accession="MS:1000544" name="Conversion to mzML"/>
37     </processingMethod>
38   </dataProcessing>
39 </dataProcessingList>
40 <run id="Exp01" defaultInstrumentConfigurationRef="TSQ" startTimeStamp="2008-06-27T15:23:45.00035"
    defaultSourceFileRef="SF1">
41   <chromatogramList defaultDataProcessingRef="XcaliburProcessing" count="2">
42     <chromatogram index="1" id="transition_445-525" defaultArrayLength="18">
43       <cvParam cvRef="MS" accession="MS:1001473" name="selected reaction monitoring chromatogram"/>
44       <precursor>
45         <isolationWindow>
46           <cvParam cvRef="MS" accession="MS:1000827" name="isolation window target m/z" value="445.34"
              unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
47           <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset" value="0.5"
              unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
48           <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset" value="0.5"
              unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
49         </isolationWindow>
50         <activation>
51           <cvParam cvRef="MS" accession="MS:1000045" name="collision energy" value="35" unitCvRef="UO"
              unitAccession="UO:0000266" unitName="electronvolt"/>
52           <cvParam cvRef="MS" accession="MS:1000133" name="collision-induced dissociation" value=""/>
53         </activation>
54       </precursor>
55       <product>
56         <isolationWindow>
57           <cvParam cvRef="MS" accession="MS:1000827" name="isolation window target m/z" value="525.13"
              unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
58           <cvParam cvRef="MS" accession="MS:1000828" name="isolation window lower offset" value="0.5"
              unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
59           <cvParam cvRef="MS" accession="MS:1000829" name="isolation window upper offset" value="0.5"
              unitCvRef="MS" unitAccession="MS:1000040" unitName="m/z"/>
60         </isolationWindow>
61       </product>
62     <binaryDataArrayList count="2">
63       <binaryDataArray encodedLength="96">
64         <cvParam cvRef="MS" accession="MS:1000595" name="time array" unitCvRef="UO" unitAccession="
              UO:0000010" unitName="second"/>
65         <cvParam cvRef="MS" accession="MS:1000576" name="no compression"/>
66         <cvParam cvRef="MS" accession="MS:1000521" name="32-bit float"/>
67         <binary>fZ/QQ0wn2UMpzOFDqBbpQw5VDkSDU...</binary>
68       </binaryDataArray>
69       <binaryDataArray encodedLength="96">
70         <cvParam cvRef="MS" accession="MS:1000515" name="intensity array" unitCvRef="MS" unitAccession="

```

```

70     = "MS:1000131" unitName="number of counts" />
71     <cvParam cvRef="MS" accession="MS:1000576" name="no compression" />
72     <cvParam cvRef="MS" accession="MS:1000521" name="32-bit float" />
73     <binary>AKDIRACAn0QAAIhEAEBYRACaYUQAg...</binary>
74     </binaryDataArray>
75     </binaryDataArrayList>
76     </chromatogram>
77     <chromatogram index="2" id="transition_445-672" defaultArrayLength="18">
78     <cvParam cvRef="MS" accession="MS:1001473" name="selected reaction monitoring chromatogram" />
79     <....>
80     </chromatogram>
81     </chromatogramList>
82 </run>
</mzML>

```

Listing 5.3: MRM_example_1.1.0.mzML

```

<!-- Line 42 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content[openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/data[at0001]
/events[at0002] [1]
/data[at0003]
/items[openEHR-EHR-CLUSTER.mass_chromatogram.v1]
/items[at0001] (Chromatogram type) = "selected reaction monitoring chromatogram"

<!-- Lines 43 - 53 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content[openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/data[at0002]
/events[at0003] [1]
/state[at0006]
/items[at0161] (Chromatogram settings)
/items[at0162] (Precursor)
/items[at0010] (Isolation Window)
/items[at0011 and name/value='Isolation window attribute']
/items[at0012] (Isolation window target m/z) = "445.34 m/z"
/items[at0030] (Activation)
/items[at0031 and name/value='Dissociation method'] = "collision-induced dissociation"
/items[at0045 and name/value='Precursor activation attribute']
/items[at0048] (Collision energy) = "35 electron volt"

<!-- Lines 54 - 60 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content[openEHR-EHR-OBSERVATION.mass_spectrometry.v2 and name/value='Mass spectrometry']
/data[at0001]
/events[at0002] [1]
/state[at0006]
/items[at0161] (Chromatogram settings)
/items[at0164] (Product)
/items[at0115] (Isolation Window)
/items[at0116 and name/value='Isolation window attribute']
/items[at0117] (Isolation window target m/z) = "525.13 m/z"
/items[at0118] (Isolation window lower offset) = "0.5 m/z"
/items[at0119] (Isolation window upper offset) = "0.5 m/z"

```

Listing 5.4: Examples of XPath's to leaf nodes to instance data stored in MRM_example_1.1.0.mzML

A screenshot of the template with line number corresponding to Listing 5.3 is shown in Figure 5.2.

GelML can be found on the code repository for GelML¹⁵. One instance file recording a 1-dimensional electrophoresis (“Gel1D_Indirect_Newcastle_v4.xml”) and one instance file for recording a 2-dimensional electrophoresis experiment (“Aberdeen_gel2d_v4.xml”) were selected for evaluation. These instance files are also provided on the attached CD.

An example template based on gel archetypes to record a 1-d and a 2-d electrophoresis experiment are shown in Figure 5.3 and Figure 5.4, respectively. These template files are also provided on the attached CD.

The following listings demonstrate the mapping of specific data recorded in the GelML instance files to the leaf nodes of the archetypes used in the corresponding template, using XPath.

Listing 5.5 shows an extract from the “Gel1D_Indirect_Newcastle_v4.xml” instance file and Listing 5.6 maps some of these instances to leaf nodes in the “1D Gel Experiment.oet” template.

Listing 5.7 shows an extract from the “Aberdeen_gel2d_v4.xml” instance file and Listing 5.8 maps some of these instances to leaf nodes in the “2D Gel Experiment.oet” template.

1D gel electrophoresis

```

1 <?xml version = "1.0" encoding = "UTF-8"?>
2 <GelML >
3   ...
4   <GelMLProtocolCollection>
5     ...
6     <ElectrophoresisProtocol name="Electrophoresis protocol (enter optional other name here)"
7       identifier="ex001:ElectrophoresisProtocol0">
8       ...
9       <ElectrophoresisStep identifier="ex001:ElectrophoresisStep0" actionOrdinal="1">
10         <ElectricalParameter identifier="ex001:ElectricalParameter0">
11           <fuge:AtomicValue value="0.01">
12             <fuge:_unit OntologyTerm.ref="ex001:OntologyIndividual9" />
13           </fuge:AtomicValue>
14         </ElectricalParameter>
15         <Duration identifier="ex001:Duration0">
16           <fuge:AtomicValue value="20">
17             <fuge:_unit OntologyTerm.ref="ex001:OntologyIndividual10" />
18           </fuge:AtomicValue>
19         </Duration>
20       </ElectrophoresisStep>
21     </ElectrophoresisProtocol>
22   </GelMLProtocolCollection>
23   <Gel1DExperiment identifier="ex001:Gel1DExperiment0">
24     ...
25     <_electrophoresisType OntologyTerm.ref="ex001:OntologyIndividual6"></_electrophoresisType>
26     ...
27     <ElectrophoresisApplication identifier="ex001:ElectrophoresisApplication0"
28       ElectrophoresisProtocol.ref="ex001:ElectrophoresisProtocol0" Gel_ref="ex001:Gel0"></
29       ElectrophoresisApplication>
30   </Gel1DExperiment>
31   <fuge:OntologyCollection>
32     ...
33     <fuge:OntologyIndividual identifier="ex001:OntologyIndividual6" term="one dimensional gel
34       electrophoresis" termAccession="sep:00150" OntologySource.ref="ex001:OntologySource1" />

```

¹⁵GelML, <http://code.google.com/p/gelml/source/browse/trunk/Examples/?r=31>

```

33 <fuge:OntologyIndividual identifier="ex001:OntologyIndividual9" term="ampere" termAccession="
    UO:0000011" OntologySource_ref="ex001:OntologySource2"/>
34 <fuge:OntologyIndividual identifier="ex001:OntologyIndividual10" term="minute" termAccession="
    UO:0000031" OntologySource_ref="ex001:OntologySource2"/>
35 ...
36 </fuge:OntologyCollection>
37 </GelML>

```

Listing 5.5: Extract from “Gel1D_Indirect_Newcastle_v4.xml” file

```

<!-- Line 6 -->
[openEHR-EHR-COMPOSITION.report.v1]
  /content[openEHR-EHR-OBSERVATION.gel-electrophoresis-experiment.v1 and name/value='Gel
    electrophoresis experiment']
  /protocol[at0013]
    /items[at0014 and name/value='Gel electrophoresis Protocol']
      /items[at0016]
        /items[at0030 and name/value='Dimension Action']
          /items[openEHR-EHR-CLUSTER.electrophoresis.v1]
            /items[at0030] (Protocol Name) = "Electrophoresis protocol (enter optional other name
              here)"

<!-- Lines 8 - 19 -->
[openEHR-EHR-COMPOSITION.report.v1]
  /content[openEHR-EHR-OBSERVATION.gel-electrophoresis-experiment.v1 and name/value='Gel
    electrophoresis experiment']
  /protocol[at0013]
    /items[at0014 and name/value='Gel electrophoresis Protocol']
      /items[at0016]
        /items[at0030 and name/value='Dimension Action']
          /items[openEHR-EHR-CLUSTER.electrophoresis.v1]
            /items[at0012 and name/value='Electrophoresis Steps']
              /items[at0013 and name/value='Electrophoresis Step Number'] = "1"
              /items[at0014] (Electrical Parameters)
                /items[at0015] (Electrical Parameter Value) = "0.01 ampere"
                /items[at0020] (Time Parameter) = "20 minutes"

<!-- Line 26 -->
[openEHR-EHR-COMPOSITION.report.v1]
  /content[openEHR-EHR-OBSERVATION.gel-electrophoresis-experiment.v1 and name/value='Gel
    electrophoresis experiment']
  /data[at0001]
    /events[at0002]
      /data[at0003]
        /items[at0006] (Electrophoresis type) = "one dimensional gel electrophoresis"

```

Listing 5.6: Examples of XPaths to leaf nodes to instance data stored in “Gel1D_Indirect_Newcastle_v4.xml”

A screenshot the template with line number corresponding to Listing 5.5 is shown in Figure 5.3.

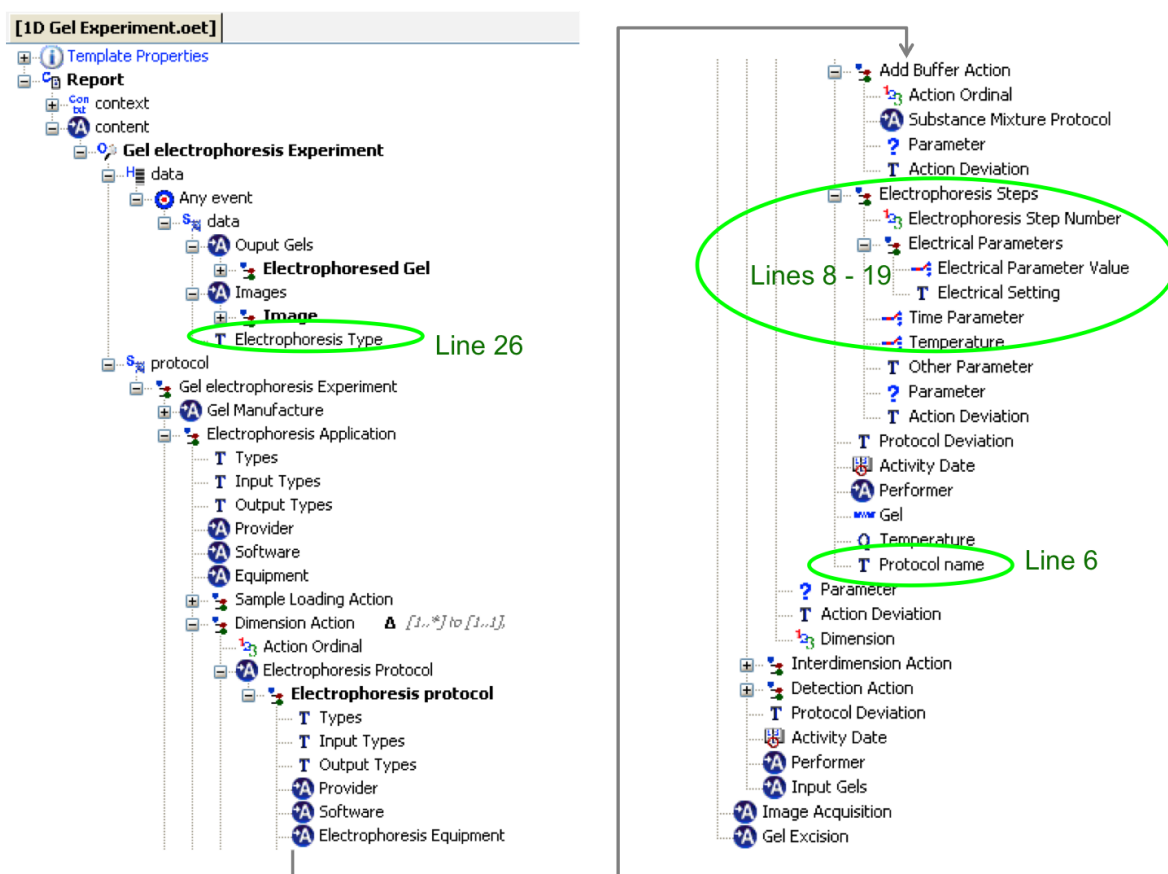


Figure 5.3: Screenshot of a template for one-dimensional electrophoresis; line numbers refer to Listing 5.5.

2D gel electrophoresis

```

1 <?xml version = "1.0" encoding = "UTF-8"?>
2 <GelML ...>
3 ...
4 <GelMLProtocolCollection>
5   <fuge:GenericEquipment name="Gels and reagents contained in 14 L Sealfresh food storage boxes,
6     approx. dimensions 290 x 290 x 160 mm (Stewart). Universal shaker SM-30, with to-and-fro
7     motion (Edmund Buhler)." identifier="ex002:GenericEquipment1"/>
8   ...
9   <DetectionProtocol name="Colloidal Coomassie Blue staining" identifier="ex002:DetectionProtocol0"
10     protocolText="Fixing: 2-3 hours with continuous shaking at ambient temperature, followed
11     by 3 x 30-60 min washes in ultrapure water. Fixative: methanol (30%), phosphoric acid (2%)"
12     >
13     <SubstanceAction identifier="ex002:SubstanceAction0" actionOrdinal="1"
14       actionText="Equilibrate with continuous shaking (80 cycles/min) for 1 h then add Coomassie
15       blue directly to solution and continue shaking for 4 days.">
16       <substanceType OntologyTerm_ref="ex002:OntologyIndividual2"/>
17       <RelativeVolume identifier="ex002:RelativeVolume0">
18         <fuge:AtomicValue value="0.067"/>
19       </RelativeVolume>
20     </SubstanceAction>
21     <_detectionEquipment GenericEquipment_ref="ex002:GenericEquipment1"/>
22     <AddBufferAction identifier="ex002:AddBufferAction2" actionOrdinal="2"
23       SubstanceMixtureProtocol_ref="ex002:SubstanceMixtureProtocol0"/>
24   </DetectionProtocol>
25   <Gel2DProtocol name="Enter a name for the Gel2D protocol here" identifier="ex002:Gel2DProtocol0"
26     >
27     ...
28   </Gel2DProtocol>
29   ...
30   <SubstanceMixtureProtocol identifier="ex002:SubstanceMixtureProtocol0" mixtureName="
31     Equilibration buffer">

```

```

23     <SubstanceAction identifier="ex002:SubstanceAction1" substanceName="methanol">
24       <RelativeVolume identifier="ex002:RelativeVolume1">
25         <fuge:AtomicValue value="34"/>
26       </RelativeVolume>
27     </SubstanceAction>
28     <SubstanceAction identifier="ex002:SubstanceAction2" substanceName="ammonium sulphate">
29       <RelativeVolume identifier="ex002:RelativeVolume2">
30         <fuge:AtomicValue value="17"/>
31       </RelativeVolume>
32     </SubstanceAction>
33     <SubstanceAction identifier="ex002:SubstanceAction3" substanceName="phosphoric acid">
34       <RelativeVolume identifier="ex002:RelativeVolume3">
35         <fuge:AtomicValue value="2"/>
36       </RelativeVolume>
37     </SubstanceAction>
38   </SubstanceMixtureProtocol>
39 </GelMLProtocolCollection>
40 <Gel2DExperiment name="Optional name for this 2D Gel experiment" identifier="
41   ex002:Gel2DExperiment0">
42   <DirectDetection identifier="ex002:DirectDetection0" DetectionProtocol_ref="
43     ex002:DetectionProtocol0"/>
44   ...
45   <Gel2DApplication identifier="ex002:Gel2DApplication0" activityDate="2003-01-20T19:00:00"
46     Gel2DProtocol_ref="ex002:Gel2DProtocol0">
47     ...
48     <Gel2D identifier="ex002:Gel2D0">
49       <PHRange dimension="1">
50         <fuge:Range lowerLimit="4" upperLimit="7"/>
51         <_rangeType OntologyTerm_ref="ex002:OntologyIndividual16"/>
52       </PHRange>
53       <MolWeightRange dimension="2">
54         <fuge:Range upperLimit="120" lowerLimit="12"/>
55         <fuge:_unit OntologyTerm_ref="ex002:OntologyIndividual4"/>
56       </MolWeightRange>
57     </Gel2D>
58     <_inputFirstDimension>
59       ...
60     </_inputFirstDimension>
61     <_inputSecondDimension>
62       <Gel name="Slab gel, denaturing, single separation zone, bonded to polyester film backing"
63         identifier="ex002:Gel1" separationDimension="2">
64         ...
65         <_percentAcrylamide>
66           <fuge:AtomicValue value="12.5"/>
67         </_percentAcrylamide>
68         <AcrylamideToCrossLinker acrylamide="97" crossLinker="3">
69           ...
70         </AcrylamideToCrossLinker>
71         <_model OntologyTerm_ref="ex002:OntologyIndividual15"/>
72         <Dimensions z="1" y="196" x="255">
73           <_dimensionUnit OntologyTerm_ref="ex002:OntologyIndividual3"/>
74         </Dimensions>
75       </Gel>
76     </_inputSecondDimension>
77   </Gel2DApplication>
78   ...
79 </Gel2DExperiment>
80 <fuge:OntologyCollection>
81   <fuge:OntologyIndividual identifier="ex002:OntologyIndividual2" term="Coomassie blue"
82     termAccession="sep:00114" OntologySource_ref="ex002:OntologySource1"/>
83   <fuge:OntologyIndividual identifier="ex002:OntologyIndividual3" term="millimeter" termAccession=
84     "UO:0000016" OntologySource_ref="ex002:OntologySource2"/>
85   <fuge:OntologyIndividual identifier="ex002:OntologyIndividual4" term="kilodalton" termAccession=
86     "UO:0000222" OntologySource_ref="ex002:OntologySource2"/>
87   <fuge:OntologyIndividual identifier="ex002:OntologyIndividual13" term="immobiline dry strip"
88     termAccession="sep:00131" OntologySource_ref="ex002:OntologySource1"/>
89   <fuge:OntologyIndividual identifier="ex002:OntologyIndividual15" term="DALT Gel 12.5"
90     termAccession="DALT Gel 12.5" OntologySource_ref="http://www.amershambiosciences.com/">
91   <fuge:OntologyIndividual identifier="ex002:OntologyIndividual16" term="logarithmic distribution"
92     termAccession="sep:00019" OntologySource_ref="ex002:OntologySource1"/>
93   <fuge:OntologySource name="PSI-SEP CV" identifier="ex002:OntologySource1" ontologyURI="http://
94     psidev.sourceforge.net/gps/CV/sep.obo"/>
95   <fuge:OntologySource ontologyURI="http://obo.cvs.sourceforge.net/obo/obo/ontology/phenotype/unit
96     .obo?view=log" identifier="ex002:OntologySource2" name="OBO-unit"/>
97 </fuge:OntologyCollection>
98 </GelML>

```

Listing 5.7: Extract from “Aberdeen_gel2d.v4.xml” file

```

<!-- Lines 7 - 17 & 22 - 38 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content[openEHR-EHR-OBSERVATION.gel.electrophoresis.experiment.v1 and name/value='Gel
  electrophoresis experiment']
/protocol[at0013]
/items[at0014 and name/value='Gel electrophoresis Protocol']
/items[at0016] (Electrophoresis Application)
/items[at0042 and name/value='Detection Action']
/items[openEHR-EHR-CLUSTER.molecule.detection.v1]
/items[at0045] (Protocol Name) = "Colloidal Coomassie Blue staining"
/items[at0039 and name/value='Protocol Text'] = "Fixing: 2-3 hours with continuous
  shaking at ambient temperature, followed by 3 x 30-60 min washes in ultrapure
  water. Fixative: methanol (30%), phosphoric acid (2%)"
/items[at0007 and name/value='Substance Action']
/items[at0008] (Action Ordinal) = "1"
/items[at0012] (Action Text) = "Equilibrate with continuous shaking (80 cycles/min)
  for 1 h then add Coomassie blue directly to solution and continue shaking for 4
  days."
/items[at0014] (Substance Type) = "Coomassie blue"
/items[at0019] (Relative Volume) = "0.067"
/items[openEHR-EHR-CLUSTER.device.v1 and name/value='Device']
/items[at0001] (Device name) = "Gels and reagents contained in 14 L Sealfresh food
  storage boxes, approx. dimensions 290 x 290 x 160 mm (Stewart). Universal
  shaker SM-30, with to-and-fro motion (Edmund Buhler)."
/items[at0031] (Add Buffer Action)
/items[at0032] (Action Ordinal) = "2"
/items[openEHR-EHR-CLUSTER.substance.mixture.v1 and name/value='Substance mixture']
/items[at0042 and name/value='Mixture Name'] = "Equilibration buffer"
/items[at0007 and name/value='Substance Action']
/items[at0013] (Substance Name) = "methanol"
/items[at0022] (Relative Volume) = "34"
/items[at0007 and name/value='Substance Action']
/items[at0013] (Substance Name) = "ammonium sulphate"
/items[at0022] (Relative Volume) = "17"
/items[at0007 and name/value='Substance Action']
/items[at0013] (Substance Name) = "phosphoric acid"
/items[at0022] (Relative Volume) = "2"

<!-- Lines 60 - 75 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content[openEHR-EHR-OBSERVATION.gel.electrophoresis.experiment.v1 and name/value='Gel
  electrophoresis experiment']
/protocol[at0013]
/items[at0014 and name/value='Gel electrophoresis Protocol']
/items[at0016]
/[openEHR-EHR-CLUSTER.gel.v1 and name/value='Input Second Dimension']
/items[at0001] (Name) = "Slab gel, denaturing, single separation zone, bonded to
  polyester film backing"
/items[at0009] (Percent Acrylamide) = "12.5 %"
/items[at0010] (Acrylamide to Cross Linker)
/items[at0017] (Acrylamide to Cross LinkerRatio) = "97/3"
/items[at0019] (Model) = "DALT Gel 12.5"
/items[openEHR-EHR-CLUSTER.dimensions.v1]
/items[at0004] (Depth) = "1 millimeter"
/items[at0002] (Length) = "255 millimeter"
/items[at0003] (Breadth) = "196 millimeter"

<!-- Lines 46 - 56 -->
[openEHR-EHR-COMPOSITION.report.v1]
/content[openEHR-EHR-OBSERVATION.gel.electrophoresis.experiment.v1 and name/value='Gel
  electrophoresis experiment']
/data[at0001]
/events[at0002]
/data[at0003]
/items[openEHR-EHR-CLUSTER.electrophoresed.gel.v1 and name/value='Electrophoresed Gel']
/items[at0002 and name/value='Separation Range'] [1]
/items[at0003] (Dimension) = "1"
/items[at0004] (Separation Type) = "pH"
/items[at0011] (Range Values)
/items[at0012 and name/value='By pH'] = "4-7"
/items[at0008] (Range Type) = "logarithmic distribution"
/items[at0002 and name/value='Separation Range'] [2]
/items[at0003] (Dimension) = "2"

```



```

/items[at0004] (Separation Type) = "Molecular Weight"
/items[at0011] (Range Values)
/items[at0013 and name/value='By Molecular Weight'] = "12-120 kiloDalton"

```

Listing 5.8: Examples of XPath to leaf nodes to instance data stored in “Aberdeen_gel2d_v4.xml”

A screenshot the template with line number corresponding to Listing 5.7 is shown in Figure 5.4.

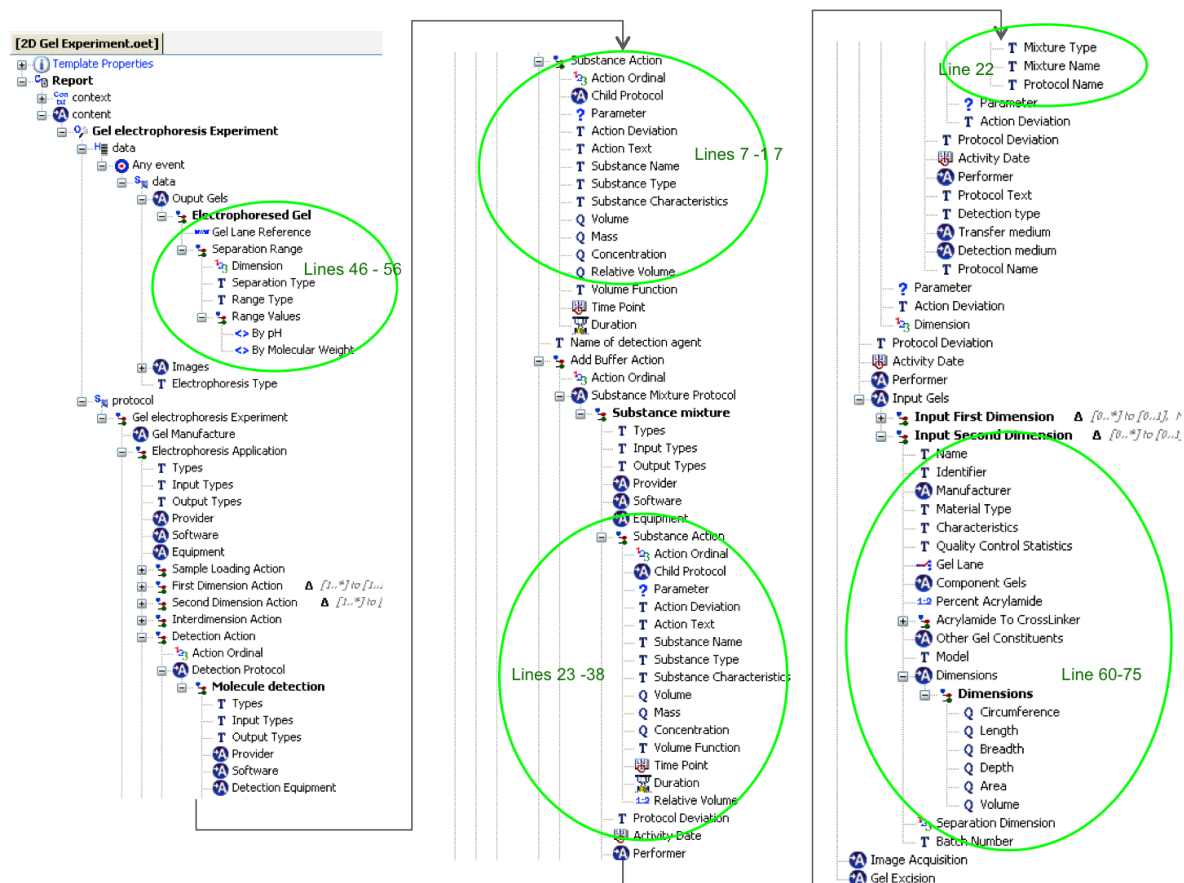


Figure 5.4: Screenshot of a template for two-dimensional electrophoresis; line numbers refer to Listing 5.7.

5.4.5 Evaluation of FuGE-based archetypes

No instance file for the generic FuGE format could be obtained. However, a minimally constrained template, including all the FuGE based archetypes developed is created to demonstrate the hierarchical organisation of the FuGE-based archetypes in a hypothetical use-case. This template file is provided in Figure 5.5 on the attached CD.

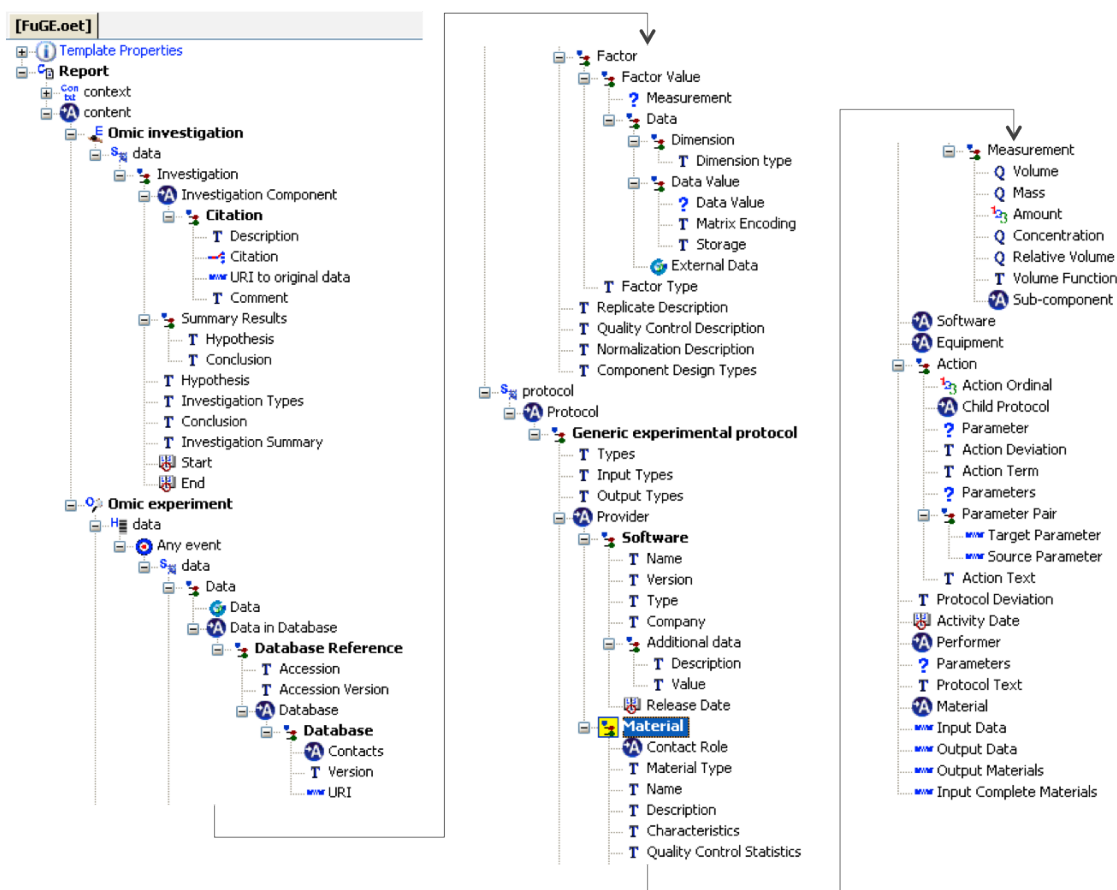


Figure 5.5: Screenshot of a minimally restricted FuGE template as shown in the Ocean Informatics Template Designer

5.4.6 Evaluation of the process used to create omic archetypes

During this research, a two-phase process to create omic archetypes from XML-based omic data formats was developed. This section provides an evaluation of the perceived usefulness of the process. As discussed in Chapter 4, both archetypes and data formats are ways of describing a meta-data structure for storing data. Both archetypes and the XML data formats are consensus-based and provide a detailed meta-data description of the information to be recorded. To model any type of data, one needs to know what data items need to be modelled, how many instances of each item may appear and how they are related to each other to create sensible information structures. Existing omic data formats provide this information for modelling omic data. It was argued that omic data formats provide a sensible basis for the development of omic archetypes because they are the output of a consensus-based approach to generically describe omic data. In addition, the XML schema provide computable and detailed definitions of the data items to be stored and their relationship to other data in an omic experiment, which could be exploited in the development of archetypes.

Review of the 2-phase omic archetype development process

Their structural similarities were exploited and a partial (syntactic) mapping could be established between the meta-information defined in the XML-based data formats and the meta-information that define archetypes. This mapping was used to automate the transformation of the meta-data definitions into syntactically valid candidate archetypes, carried out in phase 1 of the omic archetype development process, which were then reviewed and modified in phase 2 of the process to create semantically sound and reusable omic archetypes. Using existing standards based omic data formats means that the step of gathering and grouping the knowledge to be modelled about a particular omic experiment could be omitted, saving modelling time over having to gather this information from scratch. Furthermore, by automating the process of syntactic transformation, the knowledge modelled in the data formats could be brought into the computational domain of archetypes, saving time over doing this manually, as the transformation software can be applied to further data formats with minimal modification.

In addition to the elements defined in the data format, elements may also be defined in accompanying CVs and included as specified in the CVMappingRules. Thus, an instance file of the data format is always a combination of both the element definitions in the data format together with the inclusion of those elements from the CVs as dictated by the CVMappingRules. Another advantage of automating the process is that the additional items from the association with CVs and CVMappingRules would be automatically included in the archetype definitions in the correct locations, thus further reducing the effort involved in trying to include them manually. The resulting (candidate) archetypes are a product of both the `xs:complexType` element definitions in the XML schema and the mapped CV terms from the associated CVs according to the CVMappingRules.

While both archetypes and XSD schemas display similarities in how they define the information to be stored, they are based on different underlying models and differ in their view of the world. The biggest difference between the omic data formats and archetypes is that archetypes model domain concepts, whereas the data format simply provides a meta-data description of information to be stored. Archetypes combine and restrict elements of the underlying *openEHR* reference model to model a specific domain concept, XSD files combine and restrict the use of XML data types to define how particular information should be represented in an XML document, but the information in the XML schema does not have to adhere to the semantic requirements of having to correspond to a domain concept. XML schemas define the contents of XML documents that comply with an XML schema with no semantic restrictions on how the content, structure and relationship of the complex elements are defined in the schema. For

example, information does not need to be organised into hierarchical concepts, but may be defined to be recorded in any suitable way or order. Generally, related information was defined under the same `xsd:complexType` element definition and this fact was used to define the minimum granularity of the candidate archetypes. In fact, in the source XSD schemas, often `xsd:complexType` elements were defined and then reused in different locations, similarly to how archetypes are defined once and then may be reused in different situations. The freedom of XML may be disadvantageous too as it allows too many different ways of recording the same data, thus making it too complicated to use (see MAGEv1 [152, 168, 128, 135, 151]). A more restricted format makes it easier to reuse data, but may make the recording of the data more cumbersome. In contrast, the *openEHR* reference model is much more strict with regard to the content, structure and relationships between archetypes. For example, the *openEHR* reference model restricts the content and scope of the knowledge modelled in archetypes to a single, non-overlapping domain concept and further restrictions are imposed on the internal structure of knowledge modelled in an archetype, depending on the choice of archetype class that was determined to be most appropriate to model a particular concept.

So, in Phase 2, the challenge was to retrofit the knowledge that is contained and modelled in a less restrictive manner within the data formats into the classes of the much more restrictive modelling methodology defined by the *openEHR* Reference Model and specifications. During this phase, it turned out that the retrofitting of the mzML data format into meaningful and reusable archetypes was more straightforward than for FuGE or the FuGE-based format GelML. The review of archetypes based on FuGE and FuGE-based data format specification required much more work than mzML. Archetypes are inherently hierarchical in their structure and organisation of information because it is based on an object model and “object models give rise to data that is inherently hierarchical in structure” (Principle 8 in “Archetype Definitions and Principles” document [38]). Thus, if the source data format from which the candidate archetypes are being created defines mainly hierarchical or pseudo-hierarchical information structures, then it is easier from the outset to organise this data in proper archetypes.

Insights gained from applying the approach to different omic data formats

One of the reasons why candidate archetypes created from the mzML data format could be more easily retrofit into semantically valid archetypes was that the information modelled by it could be modelled with true hierarchical information structures without losing information. Any non-hierarchical, non-composite relationships created through the referencing mechanism in mzML could be resolved, mainly by removing

the container type candidate archetypes and including elements that would have been listed in those containers and then referred to from another archetype directly in that archetype. In summary, in mzML, any non-composite associations could be easily replaced with composite associations and removing the previous composite association without loss of context of the information to be modelled, so that a true archetype hierarchy without horizontal referencing could be created in the resulting archetypes.

It is much more difficult and work intensive to retrofit candidate archetypes from a source XML format that models an information web rather than an information hierarchy into the *openEHR* archetype classes to create semantically sound and reusable archetypes. Specifically, FuGE and the FuGE based data formats model information with multiple non-composite associations on top of the containment association, so that instance data is heavily interlinked, possibly resulting in circular relationships, resulting in networks of information rather than information trees. Multiple non-composite associations were particularly encountered in the complex referencing and nesting structure used to represent complex experimental protocols and their run-time application in FuGE and FuGE-based formats. Modelling these complex relationships allows for experimental protocols to be described once and then reused many times in instance files, without having to describe them again. Run-time differences to the protocols are recorded separately and are linked back to the protocol instances. Due to the much more hierarchical nature of archetypes, it was difficult to find a solution that satisfies semantic archetype design principles while representing and retaining this feature, so that in the end the candidate archetypes corresponding to the elements in the data formats affected by the multiple non-composite associations (i.e. “Protocol”, “Action”, “ProtocolApplication” and “ActionApplication”) were combined into one archetype. In other cases it was not always possible to eliminate all non-composite associations without losing context, so that the non-hierarchical, horizontal relationships needed to be preserved in the resulting archetypes. While the *openEHR* reference model allows for referencing items instantiated elsewhere in the record, this violates the semantic principle of modelling data in a hierarchical manner.

Furthermore, the level of domain specificity defined by the `xs:complexType` element definitions of the mzML data format correlated much more closely with the ontological level of knowledge to be modelled in archetypes than it did in the case of FuGE or FuGE-based formats, so that the review process in phase 2 of the omic archetype development process was much more straightforward in the case of mzML than for the FuGE and FuGE-based formats.

The mapping and resulting automatic transformation process in phase 1 of the omic archetype development process was based on the mzML data format and an assumption was taken that the `xsd:complexType` elements defined in other data formats would

be defined approximately on the same ontological level. However, upon applying the software based on the mappings generated on the mzML data format to FuGE and inspection of the resulting candidate archetypes, it was found that the ontological level of the content of many of the `xsd:complexType` elements defined in FuGE is on a more generic level than in mzML. Thus, most of the information modelled in FuGE is not domain specific and too generic to be used in archetype development and corresponds closer to the first level of *openEHR* modelling approach, the information level, rather than to the knowledge or domain level modelled with archetypes. Therefore, many of the corresponding automatically created candidate archetypes are too low-level to warrant their existence as archetypes. Many of these `xsd:complexType` elements represented a type of wrapper for data types, which in archetypes are represented directly with existing data types, for example a boolean data type for boolean data. While the overall process of basing the creation of omic archetypes on existing data formats is still valid, the consequence of a more generic level of content meant that many of the candidate archetypes for FuGE and FuGE-based data formats reviewed in phase 2 need to be eliminated because they do not represent sensible archetypes.

The process of reviewing the candidate archetypes that are automatically created from existing data formats to adhere to archetype ideology and principles proved to be a work and time-intensive process. Often, several issues need to be considered in parallel, and one structural change in one archetype may have an effect on the overall compositional hierarchies of archetypes within templates. With the trend towards basing future data formats on FuGE, the review process of phase 2 on candidate archetypes created from other FuGE-based data formats may be eased by adjusting the mapping and corresponding software functionality for FuGE and FuGE-based data formats. For example, instead of creating a candidate archetype for the `xsd:complexType` for `BooleanValueType` which describes a boolean value and creating slots for this archetype that use this boolean value archetype, an `ELEMENT` of type `DV_BOOLEAN` should automatically be created wherever this value type is being referred to instead. This would reduce the review work that currently needed to be carried out on FuGE-based data formats.

Preview into applying the omic archetype development approach to further FuGE-based data formats

To get an indication of the effort involved in applying the 2-phase archetype development process on further FuGE-based omic data formats to create omic archetypes, preliminary work was carried out on two other FuGE-based data formats, spML and MAGE. spML is another markup language developed by HUPO PSI¹⁶ as an exten-

¹⁶spML, <http://www.psdev.info/spml-milestone-2-docs>

sion to FuGE, to describe sample processing steps in proteomic experiments. The XSD schema for the spML object model is similarly structured to the GelML XSD file. Thus, the transformation software adapted previously for FuGE/GelML could be reused almost without change for spML, creating 64 candidate archetypes, corresponding to the 64 elements defined in the spML XSD file. A visual inspection of the resulting candidate archetypes showed that many of the candidate archetypes were actually the same as those created from GelML, such as those created for the elements with name “VolumeParameter”, “AbsoluteVolume”, “RelativeVolume”, “TimePoint”, etc. As explained in Chapter 2, three data formats have been created for microarray data that can be used to create MIAME-compliant documents¹⁷: MAGE-ML, which is based on a complex MAGE-OM¹⁸, the much simpler tab-delimited MAGE-TAB, which is the de facto used format, and MAGEv2, which is based on FuGE, but was never adopted by the microarray community¹⁹. The transformation software developed above can only use XML based input files. No XSD schema for the newer but unused MAGEv2 could be found or created due to missing files, so the XSD representation of the MAGE(v1)-OM was used as input for the transformation software. The software for the automatic transformation of MAGE files had to be modified to account for differences in the schema definition between mzML and FuGE. For example, in the MAGE files, placeholders for ontology terms are defined through an “OntologyEntry” type. Each package of the MAGE-OM was encoded in a separate XML schema file, providing 17 XSD files, with definitions for a total of 128 “xsd:complexType” elements, resulting in 128 candidate archetypes. A visual inspection of the candidate archetypes showed similarities to other candidate archetypes developed from FuGE-based data formats, indicating similar tasks to be carried out.

Thus, it is expected that the review process on further FuGE-based data formats, such as spML and MAGE will be similar to that needed to be carried out on GelML. However, the knowledge gained from the review processes carried out on both FuGE and GelML will help reduce the time needed for this step. The fact that spML and MAGE-ML define many of the same elements as GelML is considered a weakness of the FuGE-based data formats, since it leads to duplication of work and possibly inconsistencies between FuGE-based data formats for pieces of the same information. This is in contrast to the *openEHR* approach which promotes elements to be defined once and then be reusable in different contexts. However, omic data formats have only emerged recently and many are still evolving, so that any overlap between them

¹⁷Mapping of MIAME concepts to MAGE-OM and the MGED ontology <http://www.mged.org/Workgroups/MIAME/MIAMEv1.1-MAGE-OntologyDraft2v1.0.htm>

¹⁸MAGE-OM as a set of hyperlinked diagrams <http://www.ebi.ac.uk/microarray/doc/software/schema/MAGE/MAGE.htm>

¹⁹personal communication with Ugis Sarkans, ugis@ebi.ac.uk

may be due to immaturity of these data formats. This immaturity is also evident in the inconsistencies encountered with the ontology mapping to GelML, as described in Section 4.5.8.

This preliminary work indicates that the two-phase omic archetype development process may be applied with minimal modification to other FuGE-based omic data formats and that the semantic review process of the candidate archetypes for these formats will involve similar considerations as for GelML and thus be of similar effort. Basing the approach on existing data formats reduces the overall development time of omic archetypes because it cuts out the process of agreeing the type and manner of information to be recorded and has the added advantage that one person without the domain knowledge can transform many of these data formats into semantically sensible archetypes that may be presented to the research community for further review and possible amendment.

Alternative to creating omic archetypes

An alternative to creating omic archetypes to record the information specified by the original data formats would be to create archetypes that embed the omic data in their original formats (such as the mzML or GelML instance files), similar to the way archetypes currently embed multimedia files or how FuGE files can embed proprietary data formats or how HL7 currently embeds omic data formats. However, in this case the information within the embedded file cannot be accessed with *openEHR* based tools, but would need additional software to read and interpret the embedded files (one for each type of embedded file). This problem may be alleviated by including some meta-data description about the embedded data file in the embedding archetype. However, this solution defeats the purpose of this thesis, i.e. that both clinical data and research data are modelled in a common data model, so that both can be safely shared. Nevertheless, embedding instance files of the original data formats as data blobs could serve as an interim solution until more omic archetypes become available.

5.4.7 Limitations of using the *openEHR* archetype approach to model omic information

While omic archetypes were successfully developed using standard omic data formats as the basis for their development, the differences in their underlying design philosophies led to several conflicts, some of which were already alluded to throughout this and the previous chapter. These conflicts stem from the fact that the *openEHR* modelling approach places semantic requirements on the knowledge to be modelled, whereas data formats do not, so their design is more flexible. The information that can be

recorded and shared using each approach should be the same. However, the semantic requirements on the modelling of archetypes make it difficult to retrofit the knowledge modelled in the data formats into semantically sound archetypes. A particular characteristic of the *openEHR* archetype modelling approach is that the modelling restrictions imposed by the underlying *openEHR* reference model not only dictates a syntactic, but also a semantic hierarchy in knowledge. In addition, the *openEHR* reference model defines a set of sub-classes for the ENTRY type class, putting further semantic restrictions on how knowledge should be modelled. When modelling knowledge with the *openEHR* archetype approach, one has to decided which archetype class to choose to describe a particular concept. *openEHR* provides a decision algorithm²⁰ to help decide which class to choose to describe a particular concept, taking into account that ENTRY classes cannot be nested.

The next few sections discuss the main limitations encountered during the process of modelling omic information with the *openEHR* archetype approach, such as the limitations regarding the choice of ENTRY type sub-classes, challenges encountered around the modelling of experimental workflows and networks of information and limitations around the entity-centric, in particular patient-centric focus of the *openEHR* approach.

Limitation regarding internal structures of ENTRY classes

On the ENTRY level, the choice of archetype class to be used is between OBSERVATION, EVALUATION, INSTRUCTION, ACTION, and ADMIN_ENTRY. Each ENTRY subclass is intended to model a different class of concept in a record and each with a different internal, sometimes very specific, semantically meaningful, sub-structure to be followed when modelling a concept with the chosen ENTRY sub-class. This means that semantic restrictions are imposed on the internal organisation of data within a particular ENTRY archetype class, where meaning is provided by the different attributes of the archetype class, which are used to meaningfully structure the particular parts of a concept to be modelled.

For example, in an OBSERVATION archetype, the parts of a concept to be modelled need to be distributed into the following categories: ‘data’, and optionally ‘state’ and ‘protocol’. Knowledge modelled under the ‘data’ attribute describes the data of the observation. Further, knowledge modelled under the ‘state’ attribute describes the state of the subject during the overall observation. Finally, knowledge modelled under the ‘protocol’ attribute describes the method or instrument used to arrive at the observation [42]. Further, ‘data’ is restricted to contain an internal structure of

²⁰Decision algorithm, <http://www.openEHR.org/wiki/download/attachments/786529/Decision+algorithm.gif?version=3&modificationDate=1195620505000>

type ‘HISTORY<T: ITEM_STRUCTURE>’, which can further contain events modelled with the ‘EVENT <T: ITEM_STRUCTURE>’ class with their own ‘data’ and ‘state’ attributes to model timed events and time series of events (see *openEHR* spec for details). Therefore, during the review process for the chosen top-level ENTRY archetype, the knowledge previously modelled in a simple information hierarchy in the CLUSTER type candidate archetypes needs to be reorganised appropriately into the semantically correct internal archetype sub-structure.

For example, in the case of mass spectrometry, it was decided that the OBSERVATION class is semantically the most appropriate class to be used to model the concept of a mass spectrometry experiment. A mass spectrometry experiment is composed of a series of scans, which may initially seem to fit well into the ‘data’/‘state’ distinction of the ‘EVENT <T: ITEM_STRUCTURE>’ class. Results of a scan are modelled in the ‘data’ section and the scan settings in corresponding ‘state’ section, and the details about the mass spectrometer device are modelled under the ‘protocol’ attribute.

However, timing information to be provided for the overall start of the OBSERVATION and each EVENT within the OBSERVATION is mandatory, which is not currently done for each individual mass spectrometry scan, or deemed necessary, as it is not modelled specifically in the source data formats. The source data format models the optional recording of the start time of the overall run of the mass spectrometry experiment, which is composed of a series of scans. Thus, this does not fit well with items mandated as part of using the OBSERVATION class to model an OBSERVATION. This means that all the information about a scan and scan settings may be better modelled under the ‘data’ item, to avoid the requirement for recording a date/time for each scan. This would require only one start time to be recorded (for which a “dummy time” value may need to be created if a time stamp is not available), to satisfy the requirement, especially in the context of legacy data. However, this means that ‘state’ information is modelled under ‘data’, thus violating the semantic concepts inferred from these attributes.

In contrast, the internal structure of, for example, the EVALUATION and ADMIN_ENTRY subclasses of the ENTRY level archetype have a much less restricted internal structure, not requiring the recording of date/time information, however, using an EVALUATION or even the ADMIN_ENTRY archetype to model the results of a mass spectrometry experiment would not be semantically correct. Thus, it was found that the semantic restrictions put onto the ENTRY class by sub-classing it into the five classes, each with their own specific restrictions on internal structure often conflicted with the structure of how information may be best modelled lead to modelling conflicts. Using the ENTRY archetype without sub-classing it (as in CEN13606) would help alleviate these limitations.

Limitation regarding the modelling of experimental workflows

Another complication was encountered with regard to the *openEHR* approach requiring knowledge to be modelled in a semantically hierarchical manner and not being able to nest ENTRY types.

As highlighted in Chapter 4, FuGE distinguishes between a protocol (modelled with the “Protocol” class), which may be defined once, such as a Standard Operating Procedure (SOP) or a method in a laboratory book, and its application (modelled with “ProtocolApplication”), which may be carried out many times and can capture runtime values, such as the person who performed the protocol and the date it was carried out [129, 138].

To recap, an experimental protocol can be recorded with the “Protocol” class (such as “GenericProtocol”), which consists of a set of steps (“Actions”, such as “GenericAction”). “Actions” themselves can refer to child “Protocols”. Thus, complex laboratory procedures can be represented by virtually nesting “Protocols” and “Actions” that refer to child “Protocols” recursively [138]. An instance of a (nested) “Protocol” represents the ideal case, for example a SOP, defining the actions that should be carried out as part of this protocol. The runtime contextualisation of the defined protocol in time and relating it to material used and the person who carried it out is mirrored by a ProtocolApplication instance (e.g. GenericProtocolApplication). In addition, in an instance of FuGE, each “ProtocolApplication” is linked back to the particular “Protocol” that was applied [129, 138]. An instance of “ProtocolApplication” allows for the description of inputs and outputs of the protocol, such as specific “Material” and “Data” which can be exploited to track samples and data through experimental workflows and data-processing pipelines, for example in a series of data transformations to link the original samples to the processed results (e.g. gene expression matrices) from the original sample, to the processing of the sample, acquisition of raw data and analysis of the data created [129, 138]. The “ProtocolApplication” class also allows for the annotation of deviations from the ideal protocol at runtime. An instance of “ProtocolApplication” contains instances of a “ActionApplications”, mirroring the structure of “Actions” within “Protocols” and allowing “ActionDeviations” to be reported. Thus, for each instance of “Protocol”, a “ProtocolApplication” is instantiated and for each “Action” instance, there is a corresponding instance of “ActionApplication”.

In *openEHR*, the concept of distinguishing between an instruction that describes the ideal actions to be carried out and the description of the actual actions that have been performed exists in the form of the *openEHR* INSTRUCTION and ACTION archetype classes. In the *openEHR* specification, an instruction specifies details about activities that should be carried out, realised by instances of the INSTRUCTION class and ACTIVITY classes. Actions performed as a result of the instruction are realised

with instances of the ACTION class, which describe the steps which have actually been performed. Therefore, the same separation between a specification or protocol and the actual execution of steps in *openEHR* exists in the INSTRUCTION and ACTION classes.

Thus, it could be imagined that a FuGE “Protocol” could be mapped to an INSTRUCTION ENTRY class, and the protocol steps, represented by FuGE “Actions” could be represented as ACTIVITIES. Then, the execution of protocol steps (represented in FuGE with “ProtocolApplications”) could be represented as ACTION ENTRY class instances in *openEHR*, which could be linked back to the corresponding ACTIVITY in an INSTRUCTION through an INSTRUCTION_DETAILS instance.

In FuGE, in a hierarchy of protocols and actions (and the corresponding application of protocols and actions) describing a complex experimental protocol, the highest level protocol models high-level, coarse-grained actions, with each nested child-protocol modelling more and more fine-grained actions, with the most fine-grained individual steps performed within an experiment modelled at the leaves of the hierarchy.

However, the INSTRUCTION and ACTION type ENTRY classes are too coarse-grained to model the individual steps in an experimental protocol, for example, each ACTION instance for example requires the point in time at which the action was completed to be recorded. Thus, modelling “Protocols” and “ProtocolApplications” with INSTRUCTION archetypes and ACTION archetypes, respectively, would not be suitable because of the large range of granularity presented by the protocols and actions in the FuGE data model. The INSTRUCTION ENTRY class provides the possibility of creating computable workflows and associated functionality, intended to describe ACTIONS that need to be carried out over a larger time frame and possibly by several people, where it may be difficult to keep track of who has carried out what ACTION. Furthermore, since ENTRY archetypes cannot be nested, if ACTION archetypes were used to model all “ProtocolApplications”, then actions that are understood to be carried out as part of child-protocols would be described on the same ontological level as actions that are carried out as part of higher-level protocols and thus cross-referencing would need to be used to relate ACTIONS that describe hierarchical “ProtocolApplications”, but this violates the principle of using a hierarchical model.

Therefore, modelling experimental protocols with INSTRUCTION and ACTION archetypes could only be considered for the highest-level protocols, with the rest modelled in CLUSTER archetypes to be used within the INSTRUCTION and ACTION archetypes through slots. However, even this would not be ideal, considering that this means that the modelling of one stand-alone experiment that may be composed of several protocols would be distributed over a number of ENTRY-level archetypes.

During the modelling process, it was decided that semantically the best fit to model

a complete experiment is on the level of an ENTRY level archetype, specifically with an OBSERVATION class instance, which most closely relates to the concept that needs to be modelled, out of the five available ENTRY type classes. ENTRYs represent the smallest stand-alone concept, and information modelled with lower-level archetypes should not be considered outside the context of the ENTRY class. This fits quite well with an experiment, where the result cannot be considered stand-alone, but only in the context of the experimental protocol used to achieve that result. Using the OBSERVATION class, the result of the experiment can be modelled within the ‘data’ sections, and the details of the experimental protocol used to arrive at the result can be modelled under the ‘protocol’ attribute, as described in the previous section.

As a solution it was decided to combine the information modelled into each of the candidate archetypes for the four related FuGE-classes (“Protocol”, “Action”, “ProtocolApplication” and “ActionApplication” and their extensions) into one CLUSTER type archetype, as described in detail in Section 4.5.4. Thus, the separation of ideal protocol and run-time values is not modelled with separate archetypes, but combined within the same CLUSTER archetype for each protocol to create reusable and semantically sound archetypes.

As a consequence, the ‘protocol’ section of an OBSERVATION archetype modelling an omic experiment generally needs to model much more information and is more important than in archetypes modelling clinical OBSERVATIONS.

Limitation regarding the modelling of networks of information

While the *openEHR* reference model generally enforces a hierarchical approach to knowledge modelling, it allows for cross-referencing of items between parts of an EHR to be modelled with a DV_EHR_URI data type. In addition, the *openEHR* reference model provides a LINK class, but instances of this class should only be used between archetyped structures, i.e. objects representing complete domain concepts, for example between two instances of ENTRY class archetypes and thus is not applicable to the challenge discussed here [241].

In addition, the CLUSTER.citation archetype can be used to reference to information held elsewhere, in the same EHR or external to the EHR, and uses the DV_EHR_URI to create the reference.

Cross-referencing is extensively used in the omic data formats, especially in the FuGE and FuGE-based formats, such as GelML and MAGE-ML. Data and processes are interlinked through references, and an instance document resembles more a web or network of data than a hierarchy of data. In FuGE, the “ProtocolApplication” class and extensions reference input and output materials/data that were acted upon and outputs from one protocol application may further act as input to another protocol

application [223]. Non-composite associations are modelled in FuGE-based formats by defining an identifier for an object and then a corresponding field that is instantiated with the identifier of the particular item to be referenced. As discussed in the previous section, the amount of non-composite associations could in many cases be reduced by remodelling composite and non-composite associations during the process of semantic review, however, in some cases the need for cross-referencing could not be completely eliminated. This particularly concerns the virtual flow of materials, such as samples and data through an experimental protocol. For example, in a gel electrophoresis experiment, first a gel matrix may be manufactured (the output of the manufacture process), then a biospecimen may be loaded onto the gel (i.e. the gel is the input to the next protocol step, with the loaded gel as the output). Finally, the loaded gel is subjected to the gel electrophoresis step, which outputs the electrophoresed gel, which then acts as input to further processing steps. In GelML, the manufacture of a gel may be modelled with a `CLUSTER.material-manufacture` archetype. The sample loading is modelled with a sample loading archetype (`CLUSTER.sample-loading`), which needs to specify both the gel and the gel lanes that the sample was loaded onto. Finally, the gel electrophoresis step is modelled with an internal `CLUSTER` in the `OBSERVATION.gel-electrophoresis-experiment` archetype.

In GelML, a gel matrix has one composite association in an instance document (in one of the `XGelXApplicationType` elements (`OtherGelApplication`, `Gel1DApplication` and `Gel2DApplication`)) and non-composite associations with the `GelManufactureApplication`, `SampleLoadingApplication` and `ElectrophoresisApplication` type elements. It was decided to leave the composite association as is and to model the remaining non-composite associations with the `DV_EHR_URI`. This provides a convenient way to cross-reference between items within a record to mimic non-composite associations. Alternatively this could have been done by defining an identifier field in an archetype modelling the to-be-referenced item and then modelling a corresponding field in the archetype that needs to reference this item. The latter can then be used to record the identifier of the identified item (for example using `DV_TEXT`), similar to how non-composite associations are modelled in the original GelML data format. Modelling these relationships with the `DV_EHR_URI` data type has consequences for how a user interface needs to be constructed to support the creation of these links, which needs to be more sophisticated than simply providing input fields.

However, modelling information this way affects how the archetypes may be used, since some items are modelled by reference, rather than by value. For example, the `CLUSTER.manufacture` archetype expects that the output material is modelled somewhere else by value, so that a reference can be created in an instance file of `CLUSTER.manufacture` to the previously instantiated output material. Omic data formats

often display a high level of modelling relationships between pieces of information through cross-references, rather than by containment, requiring rather the modelling of a network of information, specifically around flow of material and data through an experimental workflow. A purely hierarchical modelling of the interrelated information of an experimental workflow with semantically valid and reusable archetypes could not be achieved. Archetypes generally represent data hierarchies, with real data values in the leaf nodes, whereas many omic XML schemas create an interlinked web-like structure, where instead of real data values, leaf node equivalents may contain references to data items instantiated by value elsewhere. In this regard, due to the higher freedom of how to structure data and the common use of the referencing mechanism in XML schemas, these are more powerful in representing complicated relationships. For example, without having to adhere to a semantic modelling hierarchy, XML can be easily used to create an information web rather than a hierarchy, which may be more useful for representing the more highly interlinked omic data. It is difficult to fit the data that is represented using a web-like ideology into a strict hierarchical model. As discussed, it is possible to model these relationships with the *openEHR* archetype approach, and the resulting archetypes make use of the *openEHR* referencing mechanism provided in the design of the omic archetypes, but doing so somewhat undermines the idea of a semantic knowledge hierarchy which the *openEHR* modelling methodology supports. The reason for this may lie in the fact that omic data from the start could be modelled with modern, computer-based methodologies, which readily and easily support cross-referencing, whereas the modelling approach of *openEHR* seems to originate from the need to model clinical documents, which are inherently more hierarchal in nature. This is apparent in the specification of the *openEHR* reference and archetype models.

Limitations regarding patient-centricity and dealing with other longitudinal entities

Another important issue that cannot be ignored if clinical and research information are to be integrated in an *openEHR* based record, is the fact that an *openEHR* based record is centred around an entity, in particular the entity of the patient, and the archetypes used to model the information to be recorded are centred around modelling patient interactions (direct and indirect).

Thus, the *openEHR* record is structured as a longitudinal, patient-centric record, with the patient the subject of care and thus the root subject of the record, the record subject around whom the record is built. In fact, the *openEHR* record prioritizes the patient/carers interaction over the research use of the record [39]. This means that every piece of information to be recorded in an *openEHR*-based record is recorded within the context of a particular patient.

However, the patient and his/her clinical background only presents part of the information that needs to be recorded in the eBMRR. As shown in this thesis, the *openEHR* approach is suitable for modelling the clinical patient part of the eBMRR, and many existing clinical archetypes can be reused for this purpose.

In fact, the subject of care in the eBMRR is the biospecimen, not the patient and in addition to the clinical information that originates from patient interactions, the eBMRR needs to support the recording of investigations and experiments on patient samples. Patient biosamples are also dealt with in a clinical context to directly support patient care, such as in pathological analysis for cancer staging and corresponding archetypes to model sample analysis in a clinical context exist. How the sample entity is currently dealt with in the *openEHR* context is described in the next section.

Modelling of the sample entity in the *openEHR*-based EHR In the *openEHR* view of the world, “all physical samples must be understood as being indirect surrogates for some aspect of the patient state at the time of sampling, which cannot be observed by direct, instantaneous means in the way a pulse can be taken” [42]. Thus, an observation on a sample, such as the microscopic analysis of the resected prostate as part of a pathological analysis, is recorded under an OBSERVATION archetype, and a description of the sample entity, e.g. the resected prostate, may be included through the provision of a sample slot for the CLUSTER.specimen archetype, which has been developed by *openEHR* to record the details of a laboratory specimen. The purpose of the CLUSTER.specimen archetype reads as follows: “To record details of a laboratory specimen. Will often be used in different contexts e.g within an Instruction archetype to describe the specimen that has to be taken, or describing the specimen which accompanies the laboratory request. It may occur within an Action archetype e.g. describing specimens taken as part of a surgical procedure. It will finally be used within a laboratory result archetype to describe the specimen being reported”. The CLUSTER.specimen archetype contains ELEMENTs to describe simple details regarding collection, handling and processing, generally through the use of ELEMENTs allowing free text (DV_TEXT).

The CLUSTER.specimen archetype also provides fields to contain the ‘Specimen identifier’ and a ‘Parent specimen identifier’, which can be used to explicitly link instances of sample appearing in the record on the data level. For example, the first occurrence of a particular sample description within a patient’s health record may be as part of an ACTION archetype describing the sample being collected and a ‘Specimen identifier’ assigned to it. Then, when the sample has been processed and sub-samples analysed by the laboratory, each OBSERVATION describing an observation on a sub-sample may contain a CLUSTER.specimen description of the sample with a new

‘Specimen identifier’ and a ‘Parent specimen identifier’ that is the same as the ‘Specimen identifier’ of the parent sample as described in the ACTION archetype detailing the sample collection. The CLUSTER.specimen archetype contains a slot for the very simple CLUSTER.specimen_preparation archetype which can be used “To record details of laboratory specimen processing”²¹. Thus, any additional actions carried out on the sample are currently modelled with the CLUSTER.specimen_preparation that is used in the CLUSTER.specimen.v1 archetype through the slot mechanism. Thus, any actions on the sample are modelled under the CLUSTER.specimen archetype, whereas any observations on a sample are modelled above the CLUSTER.specimen archetype.

Since the CLUSTER.specimen archetype is a CLUSTER archetype, it sits towards the bottom of the EHR Extract hierarchy. For example, it can only be used under an ENTRY archetype in the archetype hierarchy, such as in some laboratory OBSERVATION archetype or within another CLUSTER archetype, as in for CLUSTER.microscopy_prostate_carcinoma. Thus, when a sample is taken and a laboratory analysis performed on it, it may be described as part of OBSERVATION.lab_test, or specialisations of it, such as OBSERVATION.lab_test-blood_glucose or OBSERVATION.lab_test-urine_protein, with a slot option to contain a sample description. Furthermore, since CLUSTER.specimen is a CLUSTER archetype, it can only contain further CLUSTER type archetypes or ELEMENT type archetypes in the hierarchy. This way of modelling the sample entity is sufficient in a record prioritising immediate patient-care in a clinical environment, since the tests and analyses carried out on the sample follow long-established, well-known and stable protocols, and the emphasis lies on the result of the analysis which is used to guide further patient care. Any protocol steps to attain the particular test result are well known and carried out the same way, possibly according to an SOP, and ignored in the recording of the analysis result, with few exceptions, such as providing simple sample handling, processing and quality details. In the clinical context, these laboratory observations are common and routinely carried out tests and this set up is sufficient to describe sample details, handling, preparation and the test results. In summary, the entity sample appears ‘sporadically’ throughout the patient-centric record, as instances of the CLUSTER.specimen archetype recorded under higher-level ENTRY type archetypes.

Modelling of a sample repository Initially, the idea was that the patient-centric record could simply be augmented with the results of omic experiments and that analysis on patient samples could be recorded in the same way, for example with newly developed omic OBSERVATION type archetypes, where the sample entity and

²¹open-EHR-EHR-CLUSTER.specimen.v1 archetype from *openEHR* repository, <http://www.openehr.org/knowledge/>

any sample preparation steps could be described for example with specialised CLUSTER.sample_preparation archetypes, such as a CLUSTER.sample_preparation-mass_spectrometry etc.

It was found that this patient-centric approach does not support the interactions to be recorded on a sample in a research environment well, where the subject of care is not the patient itself, but the patient's sample(s). Simple sample management, which is an integral part of any biobank (including the PCRC biobank), is thus cumbersome to model in a patient-centric model and a more multi-nodal approach that supports multiple interlinked record subjects or entities would be more suitable.

Currently, as explained above, the sample entity is modelled through the CLUSTER.specimen archetype that is included as appropriate within ENTRY level archetypes, e.g. to describe a specimen that was taken as part of a surgical procedure with an ACTION archetype. However, this was deemed unsuitable for modelling a sample repository, since the CLUSTER.specimen archetype may be included at different paths in different ACTION archetypes and thus would need different fine-grained queries carried out, one for each ACTION archetype to find where the sample may be stored. Furthermore, to update the fact that the sample has been moved or used up, the instance of the ACTION archetype describing the surgical procedure that resulted in the sample collection would need to be updated with the new information. If this ACTION archetype is part of an event type composition, this would contradict the meaning of changes to the event type archetype - generally these would be corrections, not updates. Thus, as a solution, it was decided that a persistent type COMPOSITION could be used to model a sample repository, storing information about samples kept in the PCRC biobank with ENTRYs that contain CLUSTER.specimen descriptions. Thus, when a researcher removes a sample or if it is being thawed and refrozen, these actions on the sample may be recorded as updates of CLUSTER.specimens that are instantiated within this persistent COMPOSITION, which corresponds to the correct semantics of changes made to the record (changes made to persistent type COMPOSITION entries represent updates, whereas changes made to event type COMPOSITION entries are understood as corrections [42]). In this regard, the specimen entity can be modelled as a part of the some ENTRY archetype, such as an ADMIN_ENTRY archetype to model the sample as an administrative entity. For example, the ADMIN_ENTRY.specimen archetype was developed to act as a holder for the CLUSTER.specimen.v1 entity; this COMPOSITION can then be updated to record any non-destructive actions carried out on the sample, such as thawing, a part being removed and the rest refrozen. New COMPOSITION entries can be created for sub-samples, and related through the sample parent identifier. Observations made on the samples are then recorded using OBSERVATION archetypes including a CLUSTER.specimen description as part of

event type COMPOSITIONs.

Modelling of omic experiments on patient samples While the way of modelling the sample as a CLUSTER archetype that may be included instantiated as part of OBSERVATIONs and other ENTRY level archetypes may be sufficient in the clinical environment where well-know protocols are used to provide analysis results and thus their recording is generally ignored, researchers, especially in the omic domain have emphasised the importance of providing the details of how a particular omic result has been obtained. In particular, omic results can only be understood and interpreted correctly in the context of the experimental protocols applied and are considered meaningless without providing the context of the analysis results. Thus, in a research context, much more information needs to be provided about the sample processing than in a clinical context. Omic research is still a very new domain, and a plethora of different and often experimental analysis techniques exist and are still being developed, and includes many alternative parameters which may have an influence on the final result. In the absence of a small number of established and stable techniques, details of the protocols used to obtain the analysis results need to be recorded alongside the result.

It was found that omic data formats model the sample as an entity that acts as input to a particular application of a protocol. Chapter 4 explained how experimental protocols are modelled under the “protocol” attribute of an OBSERVATION archetype that models the complete experiment. This is inconsistent with the current way of modelling actions on samples, i.e. within the CLUSTER.specimen archetype. In the omic context, this modelling is turned “upside down”, so that actions on samples are described in the protocol section of the OBSERVATION archetype, with the sample entity acting as input to these protocols. The idea of creating a sample-based sub-record, with the sample at the centre of the record was briefly considered, but does not fit well with how researchers think in terms of investigations and experiments. Experiments are not modelled as actions and observations and evaluations on samples, but rather as applications of protocols to which a sample may be input and a sample is not an entity that is interacted with over time (in most cases the sample is destroyed during an experiment - there are exceptions, such as cancer cell lines).

The solution suggested above works well for recording experiments on the samples of one patient as it does make sense semantically to place the information into the record of the patient from which the samples originate. However, often in research, experiments are carried out on several samples combined, for example from two different patients or patient groups, to compare between them and where the results need to be interpreted relatively to each other (such as cancer patients displaying stage 1 cancer vs. cancer

patients displaying stage 2 cancer). Thus the question arises under which patient should the execution and results of these experiments be placed? The underlying reference model does not allow for this information be placed outside a record that is centred around a record subject.

To answer this question, the basis for longitudinal recording in a research environment needed to be identified. While more research on this is needed, for the interim, it is proposed that the concept of a “scientific investigation” is the longitudinally sensible entity that may be designated the root subject of a record. This record can be used to record the experiments carried out as part of this investigation in a research environment, with its own observations, evaluations, instructions and actions and administrative information associated with it. This record may exist in parallel with the patient records, so that the eBMRR system contains a collection of patient-centric and investigation-centric records. In the patient-centric *openEHR*-based EHR, any other objects, such as the patient’s samples, are not treated as subjects of information. This presents a problem when these objects are treated as entities in real life, such as the sample entity, that may exist longitudinally for some time and may need to have a mini-record to be kept about them that is directly relevant to the primary record subject. The sample entity is but one subject of information that may need to be considered as an identified entity with the need for a mini-record. It is not clear yet if this is the best solution and others may need to be investigated. It is however clear that a patient-centric, hierarchical approach such as *openEHR* has some limitations with regard to fulfilling the requirements of the proposed eBMRR.

In this section, the feasibility of using the patient-centric *openEHR*-based approach to implement the eBMRR has been discussed. This discussion lead to a re-evaluation of centring the eBMRR around the patient/donor and showed the need for a parallel record structure to record investigations involving patient samples. During the course of this research, it was found that using the patient-centric *openEHR* approach in a research context presents problems that make it cumbersome to satisfy all the requirements of the eBMRR. An alternative solution, where a record is instantiated in parallel with an investigation at the centre of the record, the root record subject, was proposed as a possible solution.

5.5 Summary of modelling omic information with *openEHR* archetypes and templates

It has been shown above that the *openEHR* approach can be used to model omic data, such as the execution and results of a mass spectrometry or gel electrophoresis

experiment. In particular, the existence of consensus-based well-defined XML-based omic data formats can be leveraged for the development of omic archetypes. Standard omic data formats already define the individual pieces of data, their occurrences and relationships to record the particular data in a way that correspondences can be established between the data definitions in the original data format and archetype definitions. These correspondences allow for the automatic transformation of the data defined in the original data format into candidate archetype structures.

However, some limitations of using the *openEHR* approach as the basis for the eBMRR have been identified and discussed. Some are related to trying to fit knowledge modelled with a more flexible model (the source data format) into a more restrictive model, the *openEHR* archetype model, which has to satisfy additional semantic requirements. Examples of these involve the choice of a particular archetype class, whose internal organisation of data definitions within archetypes conveys meaning, whereas the original data formats do not have these restrictions.

Although *openEHR* claim their reference model to be independent with regard to the domain, so that a record for any domain may be modelled with it, the semantic restrictions that it imposes on the classes that may be used to create archetypes are not always a good fit for the research domain. While the model works well to support the modelling of concepts to support the recording of patient-interactions, it works less well in the research domain to support the recording of the execution and results of experiments on (patients') biosamples. One issue surrounds the decision of the *openEHR* to categorise the concepts to be modelled into five different ENTRY categories, which are less suitable to model the world in the view of researchers, who think in terms of investigations and experiments.

The particular class of archetype used to describe information is also meaningful, so care needs to be taken to choose the correct archetype class, so that information is described at the correct ontological level. This leads to a particular problem: `xsd:complexType` elements defined in XSD files can be nested indefinitely in any way, and links between data items are easily established. In archetypes however, of the information bearing archetypes (ENTRY and below), only the CLUSTER type archetypes can be nested indefinitely. Thus, the challenge is to resolve conflicts where certain information should ontologically be described on the ENTRY level, but because it needs to be nested, it can only be described on the CLUSTER level. Furthermore, in archetypes, the use of cross-linking and referencing of described elsewhere information is generally uncommon, although not impossible. It was found that a lot of the cross linking and referencing in the original data formats could be eliminated by including the data referred to directly under the specific item, if it was related to the concept, thereby following the stricter hierarchical organisation of data in archetypes.

During the reorganisation of archetypes in the review process of candidate archetypes it was important to preserve relationships between data items. Where multiple relationships existed between items, especially in FuGE and FuGE-based data formats, it was not always possible to model knowledge in a purely hierarchal manner. In addition, it turned out that FuGE and FuGE-based data formats define a large number of generic components that were ontologically more closely related to the level 1 of the *openEHR* reference model, adding to the work to be carried out during the review process. In future, a mapping between these generic components and archetype constructs and corresponding modifications in the automatic transformation software could help to ease this process.

However, it has also been shown that the process has been successfully applied to create semantically sound omic archetypes to record the execution and results of two omic experimental techniques, mass spectrometry and gel electrophoresis, and that these archetypes can be used to faithfully record the same omic data as by instances of the omic data formats. It has been shown that the 2-phase approach to develop omic archetypes from omic data formats is useful for the creation of semantically sound omic archetypes and can easily be adapted to further omic data formats to create further omic archetypes. An advantage of this approach is that it can be carried out by one person with only basic domain knowledge and does not have the usual overheads associated with creating archetypes *de novo*, such as building balanced groups of experts and committees to discuss the concepts and exact data items to be recorded. By essentially making this step redundant, the effort to develop omic data formats is significantly reduced. The archetypes can now be exposed to a final review by domain experts. In addition, by basing them on these data formats, a conversion between data instances is going to be straightforward, since the mapping is already established.

5.6 Summary

To reiterate, this thesis set out to investigate whether the *openEHR* approach would be a feasible approach to use for the development of a generic electronic biomedical research record based on the study participant/donor for biobanks that integrates clinical and research information and can be safely shared amongst collaborating researchers. In this chapter, the various objectives set out to achieve this aim were evaluated. In particular, the feasibility of reusing existing archetypes to model the patient's clinical information in a biobank database, using the example of the PCRC database was evaluated. Since this exercise was carried out on an existing biobank, it also gave insight into the type of conflicts to expect due to modelling differences when trying to integrate information from a legacy database into an archetype-based record sys-

tem. However, the fact that the majority of the information in the database could be modelled with existing archetypes confirms the genericity of the archetype approach by proving that archetypes can be reused in a different context and acted as confirmation of the feasibility of using the *openEHR* approach as the basis for the proposed eBMRR.

The second part of the evaluation concerned the feasibility of modelling results of omic experiments on patient samples with *openEHR* archetypes. Sets of generic omic archetypes were developed, modelling the execution and results of mass spectrometry experiments, a simple omic experiment and the execution and results of gel electrophoresis experiments. The validity and usefulness of these archetypes has been evaluated. Further, it was shown that it was useful to base the development of these sets of omic archetypes on standard omic data formats, as these provide a detailed account of the information to be recorded in a computable format that could be leveraged to automate some of the process of the creation of omic archetypes. A general 2-phase approach for developing omic archetypes from standard omic data formats was defined. It was found that some data formats were easier to use as a basis for the development of omic archetypes than others and possible reasons were discussed.

While the results of the omic archetype development process were promising in confirming the feasibility of using the *openEHR* approach as basis for the eBMRR, some limitations were identified. This realisation lead to a re-evaluation of centring the eBMRR, which will be used in a research context, around the patient/donor as the root subject of the record. While more research is needed, a possible solution with an “investigation” acting as the root of the record has been proposed.

In summary, it was found that the *openEHR* approach could be used as basis for the eBMRR that integrates both clinical patient and research information on patient samples in a shareable way. Existing archetypes could be reused to record the patient’s clinical information and new omic archetypes can be developed from standard omic data formats. Further research is needed to address the identified limitations, such as the current way of modelling the other entities that need to have information recorded on them, independent of the patient, and to satisfy the increased need to model networks of information in the research domain.

The next chapter provides a summary of the insights and key contributions of this thesis and and proposes possible avenues of future work that may be advanced from this thesis.

Chapter 6

Conclusions and Future Work

6.1 Introduction

With the trend towards translational research, from bench to bedside, the boundaries between research and the provision of clinical care are going to blur and it is expected that scientific research results will in the future more quickly and more directly influence a patient's treatment. To support this development, a prerequisite is the management and integration of the different types of information, from both the clinical and the research domain. Biomedical knowledge discovery requires the ability to integrate data from two different sources: the patient's clinical background information, which is the product of both genetic and environmental influences and recorded by a clinician in the patient's medical record, and the research information produced by researchers on patient samples that are stored in biobanks, such as omic data produced from high-throughput analysis. To fulfil this need this thesis proposed the creation of a novel type of electronic record, the electronic Bio-Medical Research Record (eBMRR) that integrates different types of research information, such as omic information created by novel high-throughput technologies on patient samples with clinical patient information in the context of biobanks.

The challenge for this record is that it be developed in a generic way so that it can be easily customised to any biobank or study, where it will ease the process of setting up a new biobank information management system. A second requirement is that it is interoperable and supports the safe sharing of clinical and research data between collaborating researchers or biobanks, and allows for the future integration of data for large scale data analysis over data sets from different sources. As explained in Chapter 1, a pre-requisite for developing such a generic record is that the information to be recorded is modelled in a consistent, but flexible manner, allowing for customisations to different recording requirements while ensuring that the information can be shared safely and reliably. To address this issue in the view of the proposed eBMRR that needs to model both clinical and (omic) research information, instead of starting from scratch, the idea was to explore whether recent standard developments in the domain

of interoperable EHRs and in the domain of omic data reporting could be leveraged and combined in the development of the eBMRR.

In particular, in the EHR domain, the emergence of 2-level modelling approaches promises the realisation of interoperable EHRs that can integrate highly heterogeneous data into a single patient-centric record in a generic way to support safe data sharing across multiple care settings and contexts. It was hypothesised that this approach may also be applied to model the information in the proposed eBMRR, applying the modelling approach to model research related information.

Of the currently existing 2-level modelling approaches, the *openEHR* approach was chosen as a starting point due to the existence of freely available archetype modelling tools, a repository of already existing clinical archetypes that may be reused in the eBMRR, a full specification of how an *openEHR* system may be implemented and emerging open-source implementation efforts towards fully functioning *openEHR* systems. *OpenEHR* promises to allow the creation of a shareable, future-proof and interoperable record in a flexible and extensible way that can be customised easily to a specific context, as is currently being applied to achieve the development of interoperable EHRs.

Furthermore, it was hypothesised that emerging omic data reporting standards and data formats could be leveraged in the development of omic archetypes. Thus, part of this research investigated whether the *openEHR* archetype modelling approach is suitable to model research information as specified in recently emerging omic data reporting standards and thus extend the archetype development into the research domain.

In this context, the research question was whether it is feasible and beneficial to use the *openEHR* approach as the basis for the development of the proposed generic electronic biomedical research record that integrates clinical and research information in a shareable and customisable record. To answer this question, several objectives were defined at the start of this thesis. This chapter revisits these objectives and describes the extent to which they were achieved, relating them to the context of the overall research question. It also summarises the key contributions that this research has made to the state of the art of modelling and integrating highly heterogeneous types of information in the multi-disciplinary domain of biomedical research. Finally, this chapter concludes with a discussion of future work that may be carried forward from this thesis.

6.2 Objectives and Achievements

This research explored the feasibility of using the *openEHR* archetype methodology [17] to support the development of the electronic Bio-Medical Research Record that inte-

grates clinical and research information in a shareable and generic way. The research question was concerned with whether the *openEHR* approach can support the construction of the eBMRR as it supports the construction of a generic electronic health record by modelling the information needs of the eBMRR with archetypes.

To answer the posed research question, the objectives of this thesis were two-fold:

1. Evaluate whether clinical archetype can be reused in the research context
2. Develop archetypes to model omic concepts

6.2.1 Objective 1: Reusability of clinical archetypes in a research context

One objective was to test if existing clinical archetypes can be reused to model the clinical information about a sample donor in a biobank. As an experiment, a collaboration with the Irish Prostate Cancer Research Consortium (PCRC) was set up. The patient-centric biobank database of the PCRC, which contained information extracted from the patient's medical record, was used as a source of information to be modelled with *openEHR* archetypes. The outcome of this modelling process was that the majority of the information stored in the PCRC biobank could be modelled with existing clinical archetypes, confirming the claim of the reusability of *openEHR* archetypes in different contexts and the usefulness of the 2-level modelling approach. Thus, this objective has been fully achieved.

This finding shows that it is possible to share archetypes between the clinical and research domains. By being able to reuse existing clinical archetypes to model clinical information in the research context, modelling effort is reduced while safe sharing and interoperability with *openEHR* based records is ensured. In addition, this finding provides clear supporting evidence of the many advantages put forward for the development of EHRs. It supports the claim that they provide much better support for research than paper-based records, whether to support the selection of patients who have received a particular interaction for further study or the extraction of large sets of records for data mining, knowledge discovery and epidemiological research. In fact, the need to record clinical background information on a sample donor in a biobank exemplifies such a secondary use case, i.e. to reuse clinical background information of the sample donor for research purposes. In future, when truly interoperable EHRs have become a reality, it is expected that relevant information could be imported directly from a patient's EHR into the envisaged eBMRR (subject to consent). This modelling process also gave an insight into the difficulties to be expected when attempting to integrate legacy data into an archetype-based record, which were found to parallel

common legacy integration problems. The consequences of these problems are that integration of legacy information stored in existing databases may require substantial effort.

6.2.2 Objective 2: Development of omic archetypes

In addition to the clinical data, the eBMRR needs to support the recording of omic research data. Thus, omic concepts need to be modelled with a set of generic and customisable archetypes that can be easily adapted to a specific biobank or study. There is no equivalent common model to the *openEHR* reference model in the research domain. However, recent omic data reporting standards and accompanying omic data formats have emerged that specify in detail what data should be stored in relation to each experiment/class of experiments. Furthermore, the standards have been developed by the research community over extended periods of time in much the same way that archetypes are the result of consensus amongst clinicians. While these omic data formats are not based on a common data model, recent efforts in particular to harmonise between existing omic data standards have lead to the definition of reusable information structures, improving their suitability to be used as the basis for the development of omic archetypes. Thus, to fulfil the objective of modelling the results of omic experiments on patient samples with a set of generic archetypes that can be easily adapted to a specific biobank or study, standard omic data formats for recording sample handling and experimental results and analysis were used as the basis for the development of omic archetypes. By using existing standard data formats, parts of the archetype development process could be automated, reducing the effort to develop omic archetypes.

It has been shown that the *openEHR* approach can be applied to the research domain to model omic experiments on patient samples and sets of generic omic archetypes have been developed successfully in this research for mass spectrometry, generic omic and gel electrophoresis experiments. It has further been shown that existing omic data formats, such as mzML, FuGE and GelML form a suitable starting point for the development of omic archetypes. Thus, the objective to model omic information with *openEHR* archetypes has been achieved, resulting in a set of generic and reusable omic archetypes that can be used to record omic information, thus confirming that the *openEHR* approach can be used to model omic information.

However, several challenges were encountered during this process due to fundamental differences in how the *openEHR* approach proposes to model information, which conflicts with the characteristics of research information. For example, the process of organising the data in the archetype classes as defined by the *openEHR* reference model in a manner that conforms with the archetype ideology proved much more challenging

in the context of modelling the experimental protocol and results of omic experiments on patient samples.

Further, it was found that there are differences in how omic data formats model omic information and that some data formats not only model domain-specific information which build the basis for archetypes, but often also model information that is ontologically closer to the level of an underlying reference model. This means that even though these data formats proved to be useful in the development of omic archetypes, the amount of domain-level information provided by these was small in comparison to the information they modelled on the level of a reference model, which needed to be eliminated first.

In addition, it was found that the archetype classes defined by the *openEHR* reference model [39], do not always provide a natural fit to model the execution and result(s) of an omic experiment. Rather, the information to be modelled is distributed somewhat arbitrarily amongst ENTRY type archetype classes and archetype class attributes to achieve a reasonable fit and work around the issue of mandatory information. As a result it is argued that the *openEHR* reference and archetype models are specific in the development of the type of longitudinal record they support. While the general notions of administrative information, actions, instructions, observations and evaluations can also be applied to the realm of research, the specific organisation of how information is classified under these types of classes and interrelationships between information recorded in instances of archetypes adhering to these classes differs from the reference model classes that *openEHR* has defined to generically capture these notions. While information to be recorded in a clinical context about healthcare interactions with the patient can easily be categorised into the five entry classes defined in *openEHR* [39], the fit is not as good for recording administrative information, actions, instructions, observations and evaluations on patient samples.

Furthermore, it was found that the *openEHR* approach has limitations with regard to the modelling of experimental protocols and workflows. While the *openEHR* archetype model provides classes that model the distinction between protocols (INSTRUCTION) and the application of protocols (ACTION), these model concepts at the incorrect semantic level. In particular, these concepts for an omic experiment need to be modelled under the “protocol” attribute of an ENTRY level archetype, but since ENTRY level archetypes cannot be nested, this distinction cannot be maintained at the level of distinct archetypes and are thus combined into one archetype.

Further, it has been found that omic data formats contain networks of information. The *openEHR* approach promotes a hierarchical modelling approach. While it is possible to model cross-hierarchy relationships with the *openEHR* reference model, it challenges the hierarchical modelling approach.

The current set of ENTRY type archetype classes may be sufficient to record the healthcare interactions with human subjects, but it appears that they are too restrictive to adequately model the large variety of omic experiments and their results and that either a more flexible set or a wider range of classes may need to be defined to adequately model the research content. In this regard it may have been more suitable to use the more generic EN13606 reference model, which does not subclass the ENTRY class, so that it provides greater flexibility with regard to how information is structured in this class. Alternatively, the *openEHR* reference model may be extended with an ENTRY subclass for “research” to describe research related information.

In summary, the objective to create omic archetypes that can be used to record omic experiments on patient samples has been achieved, although some limitations with the approach have been found and noted.

An additional question that arose during the research concerned how the information related to the samples in the biobank, such as the list of samples available in the biobank for a particular patient, sample processing information on the available samples, and information relating to the execution and results of an experiment on a patient’s sample(s) within and across the patient-centric record may be suitably aggregated on a high-level. For example, *openEHR* reference model assumes a record subject as the unifying factor, in the case of an EHR it is a patient, under which the information concerning this patient is recorded. The clinical *openEHR* archetypes have been designed with the patient as the record subject in mind. This was thought to fit well with the way the PCRC biobank organises its information, since it also employs a hierarchical record centred around a sample donor. While it is relatively easy to find a suitable high-level organisation for recording experiments, including raw data, on single patient samples within the patient-centric record, problems arise when an experiment is carried out on a group of samples from different patients, producing combined raw data (for example a comparison between cancer and normal tissue samples) or yielding a single result. Since this is aggregate information it does not make sense to record this information under one particular patient involved in the study.

A number of preliminary solutions and their limitations with regard to how the information may be aggregated have been discussed, including the proposal to create an investigation-centric record structure in parallel with the patient-centric records, specifically for recording results of experiments combining multiple samples from different patients. This idea would better reflect how researchers record research information as reflected in the omic data standards. However, further research is required, including the consideration of the creation of separate records for sample entities.

6.2.3 Reflection

On reflection, it may have been better to use a more generic reference model, rather than the *openEHR* approach, which seems to be restricted in the type of record they support, i.e. a longitudinal record of interactions between parties and the record subject over the time-frame of years. In general, only one record subject, the patient, is expected. Research on patient samples needs to have access to the clinical information of the patient for research (representing a secondary use case), but needs to further record research results on patient samples, which need to be interlinked to the patients from which the samples were obtained. Thus, the information modelling requirements of the eBMRR are more complex and advanced than the *openEHR* reference and archetype model currently support, such as requiring a more multi-nodal approach to support different types of record subjects and be able to group patients and other information.

This research explored the solution from an EHR perspective, trying to augment the EHR with omic information, using an underlying reference and archetype model that has been developed for a particular type of longitudinal record centred around one particular entity. However, in hindsight, maybe a solution to this problem of integrating omic and clinical information is better approached from a more generic modelling approach than the *openEHR* approach, where the modelling of clinical information is only a sub-domain of a larger set of domains that may be modelled with a more generic and multi-nodal modelling approach.

6.2.4 Conclusion

The purpose of most current biobanks is sample management, but there is now an increased drive to incorporate research data from experiments into the biobanks as well. In fact, some researchers believe that we are now on the verge of being able to combine the results of molecular analyses retrieved with high-throughput genotyping technologies with life-style and demographic data on a large scale [14]. The eBMRR proposed in this research contributes towards enabling this goal.

As a conclusion, the results of this research are promising with regard to using the *openEHR* approach as a basis for the development of the proposed eBMRR which is constructed from clinical and omic archetypes, thus allowing for research information and clinical information about a patient/donor to be integrated in a generic way in a biobank setting. It is particularly encouraging that many existing archetypes to record the patient's clinical background can be reused in the context of biobanks, thus reducing development time and ensuring that the information can be shared safely. Furthermore, it has been shown that omic archetypes can be developed and that it is beneficial to use

existing data formats as the basis for their development, with some limitations. Some challenges remain with regard to the overall organisation of information produced on the sample to support sample management and research in biobanks. Thus, to answer the research question posed in this thesis, using the *openEHR* approach as a basis for the development of the proposed eBMRR is feasible and beneficial, however, some challenges remain to be solved.

By basing the eBMRR on the *openEHR* approach the process of setting up new biobank information management systems may be eased in the future, while ensuring semantic and syntactic interoperability amongst the diverse types of information to be recorded, to support collaborating researchers and future large-scale multi-nodal data integration and analysis. This record can, once enough clinical and omic archetypes to sufficiently cover the respective domains become available, by virtue of choosing a different set of archetypes and creating templates from it, be adapted easily to different research information recording requirements and thus can be customised to a specific biobank or study. This would ease the process of setting up a new biobank or study while enabling the safe sharing and communication of biomedical information between collaborating researchers and making it available for future large-scale integrative data analysis.

6.3 Contributions to our knowledge

This section will list the contributions made to the state of the art of using the archetype approach for the development of the eBMRR.

The contributions described in Sections 6.3.1 to 6.3.4 have been published in the “International Journal of Medical Informatics” [232].

6.3.1 Reusability of clinical archetypes

In this research a set of generic archetypes was defined that can be used to model the information in the database of the PCRC biobank. Most of the information in the biobank could be modelled with existing clinical archetypes. During this research, it was found that clinical archetypes developed for the EHR context can be reused to model the clinical information on sample donors in a biobank context, thus reducing development effort of the proposed eBMRR. The identification of archetypes modelling specific desired elements of the legacy database was time-consuming, but it was promising that the majority of these archetypes could be sourced from existing repositories, validating the genericity and reusability of *openEHR* archetypes in a different, non-clinical context. It is suggested that it would be beneficial to include biomedical re-

searchers in the development of clinical archetypes to further improve their reusability in secondary, non-clinical contexts.

6.3.2 Approach for modelling data with archetypes

An process to model the data in the database of a legacy biobank database with *openEHR* archetypes was defined, containing several steps, starting with the analysis and reorganisation of the data in the database into concepts, and then searching public archetype repositories for matching archetypes. Further steps of this process depend on the existence of suitable archetypes to cover the concept that needs to be modelled, for example, a new archetype may need to be designed if no archetype could be found to cover the particular concept. Alternatively, an existing archetype may be reused as is if it fully covers the concept or may need to be modified, extended or specialised if it does not cover the concept fully. These steps are necessary because the archetype approach is still new and the number of stable archetypes is still quite small. Finally, the archetypes can be arranged into templates to customise the use of the archetypes to the requirements at hand. This process was found to be essentially the same as developed by Buck et al. [57] for modelling data in an EHR, which was published at the same time that this work was carried out and shows that the same approach used for modelling data in EHRs can be applied to modelling data in biobanks with archetypes.

6.3.3 Integration of legacy systems with archetypes

This modelling effort has also contributed valuable new insights into the challenges associated with integrating data in legacy databases into an archetype-based system. It was found that many of the challenges mirror those commonly encountered during the integration of legacy databases. Many differences have been found in the way that information is modelled in the archetypes versus how the information is modelled in legacy databases, giving rise to a number of modelling mismatches. To integrate data with archetypes, the added complication of using the archetype approach is that to fulfil the interoperability goal, one is restricted to using existing archetypes, which represent the “immutable super schemas” to which all the legacy data needs to be mapped. Due to the modelling differences, a direct mapping is often not possible, particularly when information is modelled at different granularities. This means that a complete live integration may, depending on the underlying legacy database, not be possible, and a solution where data is migrated from the legacy system into the archetype system, most likely involving human intervention, may be more suitable.

6.3.4 Archetypes and templates for the PCRC biobank

A set of 20 new, one specialised, four modified and extended archetypes, have been designed that, used together with 30 existing archetypes, can be used to model the data in a legacy BIMS, namely the BIMS of the Irish Prostate Cancer Research Consortium (PCRC) biobank. These archetypes have been organised into 18 templates, corresponding to the 18 tables provided in the legacy database. Half of the archetypes needed were reused from the *openEHR* repository.

6.3.5 Strategy for developing omic archetypes

An important contribution has been the development of a novel 2-phase process for developing omic archetypes, making use of XML-based standard omic data formats designed for sharing omic data in the research domain. In this process, the data modelled in the data format and associated ontologies has been mapped to *openEHR* reference model constructs, allowing for an automatic transformation of the data modelled in the omic data formats into the *openEHR* computational realm, thus saving modelling time. The resulting candidate archetypes underwent a review process that resulted in a set of reusable omic archetypes. The insights gained during this process will ease the development of omic archetypes from further omic data formats.

6.3.6 Development of generic omic archetypes

Using the above process, a key accomplishment has been the development of sets of reusable omic archetypes that generically model the execution and results of omic experiments carried out on patient samples. Specifically, omic archetypes have been developed recording the execution and results of mass spectrometry and gel electrophoresis experiments. These archetypes can be composed in templates to record different types of mass spectrometry and gel electrophoresis experiments by adapting them through the use of templates. The archetypes generically model sub-concepts of these experiment types and thus may also be reused in other contexts, as part of modelling the execution and results of other omic experiments. By modelling the information at the level of archetypes rather than embedding instances of the original data format in an archetype instance, information sharing between biobanks has been enhanced as both clinical and research information are modelled using the same reference model, thus allowing it to be viewed and queried in the same computational domain and thus by any archetype-enabled system. By using archetypes to record this information, information sharing between biobanks has been enhanced, since both the clinical and the research information uses a common reference model. Together, the clinical archetypes and omic archetypes for recording research results on patient samples can be used to

construct the proposed eBMRR that can easily be customised to the needs of specific biobanks.

6.3.7 Using the *open*EHR archetype approach to model non-clinical information

This research has also provided new insights into the wider applicability of using the *open*EHR archetype approach to model other information domains outside of the domain of healthcare delivery, such as omic research information. In particular, it was found that the classification of the ENTRY type classes into the five sub-types, each with a different mandated information sub-structure, may not always be a natural fit to record non-clinical information, such as experiments on patient samples. Similarly, it was found that the differentiation of persistent and event compositions cannot always be assigned in a clear-cut manner in a non-clinical context, as shown on the example of describing information related to the patients' samples, which in essence builds its own sub-record within the record. Thus, to avoid a semantic violation of the *open*EHR principles, it may be more suitable to use EN13606 archetype. EN13606 ENTRY archetype classes are more generic and allow more flexibility with regard to how information of different ENTRYs may be structured and do not need to classify the information to be stored.

6.4 Future Work

Several avenues of future work are identified from this research, which are described in the following sections.

6.4.1 Consultation with research scientists

One of the first steps should be to introduce research scientists to the concept of modelling research information with archetypes and to consult with them about the omic archetypes developed in this research. This consultation will allow further refinement of the omic archetypes which have been created.

6.4.2 Creation of archetypes from further omic data formats

This research has shown that omic XML-based data formats can be used as a starting point for the creation of omic archetypes. Two omic data formats, the mzML and GelML data formats for recording mass spectrometry and gel electrophoresis experiments and results respectively, were used as basis for the development of omic

archetypes. However, the data covered by the data formats and corresponding archetypes only represent a small snapshot of the experiments carried out in an omic investigation pipeline. For example, gel electrophoresis is only one of many possible purification techniques that may be used to prepare a sample for mass spectrometry analysis. Thus, to be able to record a complete workflow, further omic archetypes would need to be developed to be able to describe the other processes. Many omic data formats are available to which the 2-phase omic archetype development process may be applied, thus easing the development of omic archetypes. In addition, it was found that the omic data formats overlap in some of their definitions, and thus with the omic archetypes developed during this research. This means that these omic archetypes may be reusable directly or may need to be enhanced according to the data recording requirements modelled in the omic data format. As an interim solution, instances of the original data formats can be embedded in archetypes, until archetypes explicitly modelling the information in the data formats have been developed.

6.4.3 Investigation on extending the reference model

As briefly discussed in Section 6.2, challenges remain with regard to integrating research information within an *openEHR* archetype-based record system. In particular, the current set of archetype classes as defined by *openEHR* are cumbersome for modelling omic research and experimental workflow information while conforming to the meaning of the particular classes and attributes. Thus, further research is required to investigate if and how the current reference model may be extended to provide classes that are more suitable for the development of archetypes for research information. Possible ideas involve the development of a more generic set of classes of which the current set builds a sub-set or of a parallel set of classes that represent general concepts of scientific research, such as the high-level concepts defined by the RSBI in the Investigation/Study/Assay (ISA) model presented in Chapter 2, that can then be archetyped.

6.4.4 High-level organisation of the eBMRR

Further research is also required on the development of record structures based on a different record subject to record aggregate research information, such as results on experiments carried out on multiple groups of patient samples, that may be used in addition or in parallel to the patient-centric record. Preliminary ideas have been provided. Future research should include investigating basing the record structure of the research-centric record on the ISA model presented in Chapter 2, with the record subject representing an investigation around which the data to be recorded is centred. In addition, research into the possible extension of the *openEHR* reference

model towards a more generic and multi-nodal model is required. In relation to this, the modelling of the eBMRR extensions to support biobank specific activities, such as sample management and study management should be considered, modelled on how an *openEHR*-based record system can support patient management.

6.4.5 Integration of ontologies and CVs

Some of the terms extracted and incorporated into the definition of omic archetypes as controlled by the CVMappingsRules file do not represent a part of the omic concept to be modelled, but represent possible run-time values. These are currently hard-coded from the version that was available at the time of conversion of mzML into archetypes. However, in future supporting software could be developed to select and display possible terms in run-time, using the latest ontology files available and following the set out mapping rules, similar to how it is done in a “Diagnosis” field with medical terminologies in the archetype EVALUATION.problem-diagnosis as shown in the code snippet in Listing 6.1. Omic ontologies are constantly updated, and by using them at run-time it is ensured that newly defined terms are available.

```

definition
  .....
  ELEMENT[at0002.1] matches { — Diagnosis
    value matches {
      DV.CODED.TEXT matches {
        defining_code matches {[ac0.1]} — *Any term that 'is_a' diagnosis(
          en)
        }
      }
    }
  }
}

ontology
  .....
  constraint_definitions = <
    [ "en" ] = <
      items = <
        [ "ac0.1" ] = <
          text = <"Any term that 'is_a' diagnosis">
          description = <"Any term that is a diagnosis in an accepted terminology">
        >
      >
    >
  >

```

Listing 6.1: Runtime integration of ontology terms

The purpose of separating the data formats from ontologies is that data formats can be kept stable, while ontologies evolve to take in new items. Thus, in the future it would

make more sense not to extract the terms and include them in the archetype definition, but to use the above structure, which allows the use of the most recent ontology files. This however works only for terms to be used as values in a `DV_CODED_TEXT` element, but not for the definition of items that can contain a value, since these form part of an archetype definition.

6.4.6 Development of device plug-ins

A lot of omic data is produced and evaluated by machine or software, for example the raw data in a mass spectrometry experiment, therefore it would be useful to provide plug-ins that can output the data from the machines automatically as archetype instance data and that can read in the archetype instance data for further evaluation. Often, data of these machines is output in proprietary format - in these cases, it may make sense to develop converters between the proprietary format and archetype format. Similarly, the record may be incorporated into workflows and electronic lab-books to support the population of the record at the point of “care”.

6.5 Summary

This thesis proposed a novel electronic Bio-Medical Research Record which allows both clinical and omic information to be modelled together using the *openEHR* approach. By basing the data modelling of this record on the archetype approach proposed by *openEHR*, heterogeneous types of information can be generically integrated and shared safely in a syntactically and semantically interoperable manner. It has been shown that the *openEHR* archetype approach is applicable to the research domain to model patient data and can be extended with omic archetypes to model omic experiments on patients samples. This allows for both clinical and research information to be modelled with the same approach, allowing for knowledge level integration and semantic interoperability. While some limitations of using the *openEHR* reference model have been detected and some challenges remain to be solved, in the future, such a record could be largely populated automatically by importing clinical data from a patient’s EHR and then augmented with omic data from experiments carried out on patients’ samples in the research environment. This would facilitate much greater sharing of information of patient and research data between collaborating researchers and allow for large scale, integrative data analysis between different research institutions. It would also allow for this information to be preserved in a future-proof manner. Information integration and sharing is a prerequisite to increase the speed and efficiency at which biomedical research can be carried out and the proposed record would thus contribute

to accelerating the knowledge discovery pipeline. Ultimately this would lead to improved healthcare, thus bringing us closer to the aim of translational medicine and personalised healthcare.

References

- [1] ISO TC 215/SC N. ISO/WD 18308 Requirements for an Electronic Health Record Reference Architecture, 2004. ISO 18308:2004.
- [2] Jane Grimson. Delivering the electronic healthcare record for the 21st century. *International Journal of Medical Informatics*, 64:111–127, Feb 2001.
- [3] Peter H J Riegman, Manuel M Morente, Fay Betsou, et al. Biobanking for better healthcare. *Molecular Oncology*, 2(3):213–22, Oct 2008.
- [4] Zachary Zimmerman, Michael Swenson, and Brock Reeve. Biobanks: Accelerating molecular medicine - Challenges Facing the Global Biobanking Community, Nov 2004. A white paper sponsored by IBM. Available online at www.msp.sbm.u.ac.ir/payeh/Genetic/Documents/biobanks.pdf [last accessed July 30, 2012].
- [5] Dean Troyer. Biorepository standards and protocols for collecting, processing, and storing human tissues. *Methods Mol Biol*, 441:193–220, Jan 2008.
- [6] Sambit K Mohanty, Amita T Mistry, Waqas Amin, et al. The development and deployment of Common Data Elements for tissue banks for translational research in cancer - an emerging standard based approach for the Mesothelioma Virtual Tissue Bank. *BMC Cancer*, 8:91, Jan 2008.
- [7] Waqas Amin, Anil V Parwani, Linda Schmandt, et al. National Mesothelioma: Virtual Bank A standard based biospecimen and clinical data resource to enhance translational research. *BMC Cancer*, 8:236:e1–10, Jan 2008.
- [8] Rivka Ravid. Biobanks for biomarkers in neurological disorders: the Da Vinci bridge for optimal clinico-pathological connection. *J Neurol Sci*, 283(1-2):119–26, Aug 2009.
- [9] George Davey Smith, Shah Ebrahim, Sarah Lewis, et al. Genetic epidemiology and public health: hope, hype, and future prospects. *Lancet*, 366(9495):1484–98, Dec 2005.
- [10] Isabelle Hirtzlin, Christine Dubreuil, Nathalie Préaubert, et al. An empirical survey on biobanking of human genetic material and data in six EU countries. *European Journal of Human Genetics*, 11:475–488, Jan 2003.

- [11] Satyajit Rohan Jayasinghe, Akshay Mishra, Angela Van Daal, and Edmond Kwan. Genetics and cardiovascular disease: Design and development of a DNA biobank. *Exp Clin Cardiol*, 14(3):33–7, Jan 2010.
- [12] P Founti, F Topouzis, L van Koolwijk, et al. Biobanks and the importance of detailed phenotyping: a case study—the European Glaucoma Society GlaucoGENE project. *Br J Ophthalmol*, 93(5):577–81, May 2009.
- [13] Georgios Voidonikolas, Marie-Claude Gingras, Sally Hodges, et al. Developing a Tissue Resource to Characterize the Genome of Pancreatic Cancer. *World Journal of Surgery*, 33:723–731, Mar 2009.
- [14] G Olund, P Lindqvist, and J-E Litton. BIMS: An information management system for biobanking in the 21st century. *IBM Systems Journal*, 46(1):171–182, Jan 2007.
- [15] Peter H Watson, Janet E Wilson-McManus, Rebecca O Barnes, et al. Evolutionary concepts in biobanking - the BC BioLibrary. *Journal of Translational Medicine*, 7(95):1–11, Nov 2009.
- [16] Juha Muilu, Leena Peltonen, and Jan-Eric Litton. The federated database – a basis for biobank-based post-genome studies, integrating phenome and genome data from 600 000 twin pairs in Europe. *European Journal of Human Genetics (2007)*, 15:718–723, Jan 2007.
- [17] Thomas Beale. Archetypes Constraint-based Domain Models for Future-proof Information Systems. In: *Eleventh OOPSLA workshop on behavioural semantics*, pages 1–18, Mar 2002.
- [18] A Brazma, P Hingamp, J Quackenbush, et al. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet*, 29(4):365–71, Dec 2001. 10.1038/ng1201-365.
- [19] Paul T Spellman, Michael Miller, Jason Stewart, et al. Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol*, 3(9):RESEARCH0046, Aug 2002.
- [20] Chris F Taylor, Norman W Paton, Kathryn S Lilley, et al. The minimum information about a proteomics experiment (MIAPE). *Nature Biotechnology*, 25(8):887–893, Jul 2007.
- [21] Eric Deutsch. mzML: a single, unifying data format for mass spectrometer output. *Proteomics*, 8(14):2776–7, Jul 2008.

- [22] Pelin Yilmaz, Renzo Kottmann, Dawn Field, et al. Minimum information about a marker gene sequence (MIMARKS) and minimum information about any (x) sequence (MIxS) specifications. *Nature Biotechnology*, 29(5):415–20, May 2011.
- [23] Renzo Kottmann, Tanya Gray, Sean Murphy, et al. A standard MIGS/MIMS compliant XML Schema: toward the development of the Genomic Contextual Data Markup Language (GCDML). *OMICS*, 12(2):115–21, Jun 2008.
- [24] ISO. ISO/TR 20514:2005(E) Health informatics - Electronic health record — Definition, scope, and context. *ISO TC 215*, pages 1–33, Feb 2005.
- [25] CEN/TC 251. prEN13940-1:2007 Health informatics -System of concepts to support continuity of care - Part 1: Basic concepts, May 2007.
- [26] CEN/TC 251. prEN 13940-2:2008.2, Health Informatics - Systems of concepts to support Continuity of care - Part 2: Health care process and workflow. *CEN/TC 251*, pages 1–90, Dec 2008.
- [27] ÖNORM. EN12967-3 Health Informatics-Service Architecture Part 3: Computational Viewpoint, Mar 2008.
- [28] CEN. EVS-EN 14822-1:2005, Health informatics - General purpose information components - Part 1: Overview, Apr 2006.
- [29] ASTM. ASTM E2369 - 05 Standard Specification for Continuity of Care Record (CCR), Nov 2005. Available at <http://www.astm.org/Standards/E2369.htm> [last accessed April, 2010].
- [30] CDISC. DCISC Standards & Innovations. Available online at <http://www.cdisc.org/standards> [last accessed August 30, 2012].
- [31] Didi Davis. IHE IT Infrastructure (ITI) Technical Framework, Volume 1 (ITI TF-1), Integration Profiles, Dec 2008.
- [32] IHE. About IHE. Available online at <http://www.ihe.net/About/index.cfm> [last accessed April, 2010].
- [33] HL7. HL7v3 Guide. Available online at <http://www.hl7.org/v3ballot/html/help/v3guide/v3guide.htm> [last accessed July 30, 2012].
- [34] HL7. Clinical Document Architecture Framework. Available online at <http://www.hl7.org/Library/Committees/structure/CDAMemberBallot.zip> [last accessed April, 2010].

- [35] Peter Schloeffel, Thomas Beale, George Hayworth, Sam Heard, and Heather Leslie. The relationship between CEN 13606, HL7, and *openEHR*. In: *HIC 2006 and HINZ 2006: Bridging the Digital Divide: Clinician, consumer and computer Eds: Johanna Westbrook & Joanne Callen Publisher: Health Informatics Society of Australia Ltd (HISA) Proceedings. Brunswick East, Vic.: Health Informatics Society of Australia, 2006.*, pages 24–28, Jun 2006.
- [36] CEN/TC 251. prEN 13606-1 version 2006-02-09, May 2006.
- [37] Thomas Beale. Archetypes Constraint-based Domain Models for Future-proof Information Systems, Aug 2000. Available online at http://www.openehr.org/files/publications/archetypes/archetypes_beale_web_2000.pdf [last accessed September 30, 2012].
- [38] Thomas Beale and Sam Heard. Archetype Definitions and Principles, Mar 2007. Available online at http://www.openehr.org/releases/1.0.2/architecture/am/archetype_principles.pdf [last accessed September 30, 2012].
- [39] Thomas Beale and Sam Heard. *openEHR* Architecture: Architecture Overview, Apr 2007. Available online at <http://www.openehr.org/releases/1.0.2/architecture/overview.pdf> [last accessed September 30, 2012].
- [40] Dipak Kalra. Electronic Health Record Standards. *IMIA Yearbook of Medical Informatics*, pages 136–144, Feb 2006.
- [41] Ken Smith and Dipak Kalra. Electronic health records in complementary and alternative medicine. *International Journal of Medical Informatics*, 77(9):576–88, Sep 2008.
- [42] Thomas Beale, Sam Heard, Dipak Kalra, and David Lloyd. The *openEHR* Reference Model: *openEHR* EHR Information Model, Nov 2008. Available online at http://www.openehr.org/releases/1.0.2/architecture/rm/ehr_im.pdf [last accessed October 27, 2012].
- [43] Thomas Beale. The *openEHR* Archetype Model: Archetype Object Model, Nov 2008. Available online at <http://www.openehr.org/releases/1.0.2/architecture/am/aom.pdf> [last accessed October 28, 2012].
- [44] Sam Heard, Thomas Beale, Gerard Freriks, Angelo Rossi Mori, and Ognian Pishhev. Templates and Archetypes: how do we know what we are talking about?, Jan 2003. Available online at <http://www.openehr.org/wiki/download/attachments/196620/Templates+and+Archetypes.pdf> [last accessed October 28, 2012].

- [45] *openEHR*. The *openEHR* Foundation. Available online at <http://www.openehr.org/about/foundation> [last accessed October 28, 2012].
- [46] David Ingram. The Origins of *openEHR*, 2002. Available online at <http://www.openehr.org/about/origins> [last accessed October 28, 2012].
- [47] María Meizoso García, José Luis Iglesias Allones, Diego Martínez Hernández, and María Jesús Taboada Iglesias. Semantic similarity-based alignment between clinical archetypes and SNOMED CT: an application to observations. *International Journal of Medical Informatics*, 81(8):566–78, Aug 2012.
- [48] Kyung Hoon Hwang, Kyo-Il Chung, Myung-Ae Chung, and Duckjoo Choi. Review of semantically interoperable electronic health records for ubiquitous health-care. *Healthc Inform Res*, 16(1):1–5, Mar 2010.
- [49] Thomas Beale. The *openEHR* Archetype Model: *openEHR* Archetype Profile, 2007. Available online at http://www.openehr.org/releases/1.0.2/architecture/am/openehr_archetype_profile.pdf [last accessed October 28, 2012].
- [50] Thomas Beale and Sam Heard. The *openEHR* Archetype Model: Archetype Definition Language, ADL 1.4, Dec 2008. Available online at <http://www.openehr.org/svn/specification/TRUNK/publishing/architecture/am/adl1.4.pdf> [last accessed October 28, 2012].
- [51] Chunlan Ma, Heath K. Frankel, Thomas Beale, and Sam Heard. EHR Query Language (EQL) - A Query Language for Archetype-Based Health Records. In Klaus A. Kuhn, James R. Warren, and Tze-Yun Leong, editors, *MedInfo*, volume 129 of *Studies in Health Technology and Informatics*, pages 397–401. IOS Press, 2007.
- [52] Catalina Martínez-Costa, Marcos Menárguez-Tortosa, Jesualdo Tomás Fernández-Breis, and José Alberto Maldonado. A model-driven approach for representing clinical archetypes for Semantic Web environments. *Journal of Biomedical Informatics*, 42:150–164, Aug 2009.
- [53] Leonardo Lezcano, Miguel-Angel Sicilia, and Carlos Rodríguez-Solano. Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules. *Journal of Biomedical Informatics*, 44(2):343–53, Apr 2011.
- [54] Marcos Menárguez-Tortosa and Jesualdo Tomás Fernández-Breis. Validation of the *openEHR* archetype library by using OWL reasoning. *Studies in health technology and informatics*, 169:789–93, Jan 2011.

- [55] NHS Connecting for Health. CEN/ISO 13606 Pilot Study Final Report, Dec 2007. Available at http://www.ehr.chime.ucl.ac.uk/download/attachments/3833859/NHSCFH.13606-Pilot-Final-Rpt_v1-0.pdf [last accessed April, 2010].
- [56] Heather Leslie. International developments in *openEHR* archetypes and templates. *Health Information Management Journal*, 37(1):38 – 39, 2008.
- [57] Jasmin Buck, Sebastian Garde, Christian D Kohl, and Petra Knaup-Gregori. Towards a comprehensive electronic patient record to support an innovative individual care concept for premature infants using the *openEHR* approach. *International Journal of Medical Informatics*, 78:521–531, Sep 2009.
- [58] Maria Hägglund, Rong Chen, and Sabine Koch. Modeling shared care plans using CONTsys and *openEHR* to support shared homecare of the elderly. *J Am Med Inform Assoc*, 18(1):66–9, Jan 2011.
- [59] Nadim Anani, Rong Chen, Tiago Prazeres Moreira, and Sabine Koch. *OpenEHR*-Based Representation of Guideline Compliance Data through the Example of Stroke Clinical Practice Guidelines. *Studies in health technology and informatics*, 180:487–91, Jan 2012.
- [60] Sistine A Barretto, Jim Warren, Andrew Goodchild, et al. Linking Guidelines to Electronic Health Record Design for Improved Chronic Disease Management. *AMIA Annual Symposium proceedings 2003*, pages 66–70, Aug 2003.
- [61] Sistine A Barretto, Jim Warren, and Andrew Goodchild. Designing Guideline-based Workflow-enabled Electronic Health Records. *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS 2004)*, 6(6):1–10, Jun 2004.
- [62] Rigoleta Dm Dias, Timothy W Cook, and Sergio M Freire. Modeling healthcare authorization and claim submissions using the *openEHR* dual-model approach. *BMC Medical Informatics and Decision Making*, 11:60, Jan 2011.
- [63] Sebastian Garde, Evelyn J S Hovenga, Jasmin Buck, and Petra Knaup. Expressing clinical data sets with *openEHR* archetypes: A solid basis for ubiquitous computing. *International Journal of Medical Informatics*, 76S:S334–S341, Sep 2007.
- [64] Helma van der Linden, Tony Austin, and Jan Talmon. Generic screen representations for future-proof systems, is it possible? There is more to a GUI than meets the eye. *Computer Methods and Programs in Biomedicine*, 95(3):213–226, Sep 2009.

- [65] Derek Hoy, Nicholas R Hardiker, Ian T McNicoll, Phil Westwell, and Alison Bryans. Collaborative development of clinical templates as a national resource. *International Journal of Medical Informatics*, 78S:S95–S100, Sep 2009.
- [66] Evelyn J S Hovenga, Sebastian Garde, and Sam Heard. Nursing constraint models for electronic health records: A vision for domain knowledge governance. *International Journal of Medical Informatics*, 74:886–898, Aug 2005.
- [67] S Garde, E J Hovenga, J Granz, S Foozonkhah, and S Heard. Towards a repository for managing archetypes for electronic health records. In *HIC 2006: 14th Australian Health Informatics Conference, Sydney, 20-22 August, 2006*.
- [68] Christian Dominik Kohl, Sebastian Garde, and Petra Knaup. Facilitating the *openEHR* approach – organizational structures for defining high-quality archetypes. *EHealth Beyond the Horizon: Get It There : Proceedings of MIE2008 the XXIst International Congress of the European Federation for Medical Informatics, In: Studies in Health Technology and Informatics*, 136:437–42, Aug 2008.
- [69] Sebastian Garde, Petra Knaup, Evelyn J S Hovenga, and Sam Heard. Towards Semantic Interoperability for Electronic Health Records: Domain Knowledge Governance for *openEHR* Archetypes. *Methods of Information in Medicine*, 46(3):332–343, Sep 2007.
- [70] Dipak Kalra, Archana Tapuria, Tony Austin, and Georges De Moor. Quality requirements for EHR Archetypes. *Studies in health technology and informatics*, 180:48–52, Jan 2012.
- [71] Sebastian Garde, Petra Knaup, Thilo Schuler, and Evelyn J S Hovenga. Can *openEHR* Archetypes Empower Multi-Centre Clinical Research? *Connecting Medical Informatics and Bio-Informatics: Proceedings of MIE2005 - The XIXth International Congress of the European Federation for Medical Informatics, In: Studies in Health Technology and Informatics*, 116:971–976, Aug 2005.
- [72] Erik Sundvall, Rahil Qamar, Mikael Nyström, et al. Integration of tools for binding archetypes to SNOMED CT. *BMC Medical Informatics and Decision Making*, 8 (Suppl 1)(S7):1–10, Aug 2008.
- [73] David Moner, Jose A Maldonado, Diego Bosca, et al. Archetype-Based Semantic Integration and Standardization of Clinical Data. *Conference proceedings of the 28th IEEE Engineering in Medicine and Biology Society Annual International Conference, New York City, USA, Aug 30-Sept 3, 2006*, pages 5141–4, Sep 2006.

- [74] P Lichtenstein, U De Faire, B Floderus, et al. The Swedish Twin Registry: a unique resource for clinical, epidemiological and genetic studies. *J Intern Med*, 252(3):184–205, Sep 2002.
- [75] Petter Hurlen, Knut Skifjeld, and Egil P Andersen. The basic principles of the synapses federated healthcare record server. *International Journal of Medical Informatics*, 52:123–132, Oct 1998.
- [76] Rong Chen and Gunnar O Klein. The *openEHR* Java Reference Implementation Project. *Proceedings of MEDINFO 2007, In: Studies of Health Technology and Informatics. 2007*, 129(Part 1):58–62, Aug 2007.
- [77] *openEHR* and UCL. Introduction to Opereffa. <http://opereffa.chime.ucl.ac.uk/introduction.jsf> [last accessed October 25, 2012].
- [78] Christian D Kohl, Sebastian Garde, and Petra Knaup. Facilitating secondary use of medical data by using *openEHR* archetypes. *Studies in health technology and informatics*, 160(Pt 2):1117–21, Jan 2010.
- [79] Koray Atalag, Hong Yul Yang, Ewan Tempero, and Jim Warren. Model driven development of clinical information systems using *openEHR*. *Studies in health technology and informatics*, 169:849–53, Jan 2011.
- [80] Izabella Diana Păun, Dragoş G Sauciuc, N Ovidiu Iosif, et al. Local EHR management based on *openEHR* and EN13606. *J Med Syst*, 35(4):585–90, Aug 2011.
- [81] Erik Sundvall, Mikael Nyström, Mattias Forss, et al. Graphical Overview and Navigation of Electronic Health Records in a Prototyping Environment Using Google Earth and *openEHR* Archetypes. In: KA Kuhn, JR Warren, TY Leong, eds., *Proceedings of the 12th International Health (Medical) Informatics Congress, Medinfo, Published In: Studies of Health Technology and Informatics.*, 129:1043–1047, Aug 2007.
- [82] Hajar Kashfi and Olof Torgersson. Supporting *openEHR* Java desktop application developers. *Studies in health technology and informatics*, 169:724–8, Jan 2011.
- [83] David Moner, Jose A Maldonado, C Angulo, et al. Standardization of Discharge Reports with the ISO 13606 Norm. *Conference proceedings : Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1470–3, Sep 2008.

- [84] Rong Chen, Gunnar O Klein, Erik Sundvall, Daniel Karlsson, and Hans Ahlfeldt. Archetype-based conversion of EHR content models: pilot experience with a regional EHR system. *BMC Medical Informatics and Decision Making*, 9:33:e1–13, Sep 2009.
- [85] Linda Bird, Andrew Goodchild, and Zar Zar Tun. Experiences with a Two-Level Modelling Approach to Electronic Health Records. *Journal of Research and Practice in Information Technology*, 35(2):121–138, Aug 2003.
- [86] CEN TC/215 WG1. ENV13606-1:2006(E) Health informatics - Electronic health-care record communication - Part 1: Reference Model, 2006.
- [87] Marc Cuggia, Sahar Bayat, Delphine Rossille, et al. Comparing the Apgar Score Representation in HL7 and OpenEHR Formalisms. *Studies in Health Technology and Informatics*, 250:250–254, Aug 2009.
- [88] Elizabeth S Chen, Sharad Manaktala, Indra Neil Sarkar, and Genevieve B Melton. A multi-site content analysis of social history information in clinical notes. *AMIA Annu Symp Proc*, 2011:227–36, Jan 2011.
- [89] Jesús Bisbal and Damon Berry. ARCHETYPE ALIGNMENT - A Two-level Driven Semantic Matching Approach to Interoperability in the Clinical Domain. *HEALTHINF 2009*, pages 216–221, Apr 2009.
- [90] Catalina Martínez-Costa, M Menárguez-Tortosa, and Jesualdo Tomás Fernández-Breis. Towards ISO 13606 and openEHR Archetype-Based Semantic Interoperability. *Studies in Health Technology and Informatics*, 150:260–264, Sep 2009.
- [91] David Moner, Alberto Moreno, José A Maldonado, Montserrat Robles, and Carlos Parra. Using Archetypes for Defining CDA Templates. *Studies in health technology and informatics*, 180:53–7, Jan 2012.
- [92] Catalina Martínez-Costa, Marcos Menárguez-Tortosa, and Jesualdo Tomás Fernández-Breis. An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. *Journal of Biomedical Informatics*, 43(5):736–46, Oct 2010.
- [93] Catalina Martínez Costa, Marcos Menárguez-Tortosa, and Jesualdo Tomás Fernández-Breis. Clinical data interoperability based on archetype transformation. *Journal of Biomedical Informatics*, 44(5):869–80, Oct 2011.
- [94] William Goossen, Anneke Goossen-Baremans, and Michael van der Zel. Detailed clinical models: a review. *Healthc Inform Res*, 16(4):201–14, Dec 2010.

- [95] William T F Goossen and Anneke Goossen-Baremans. Bridging the HL7 template - 13606 archetype gap with detailed clinical models. *Studies in health technology and informatics*, 160(Pt 2):932–6, Jan 2010.
- [96] Juan G Bernal, Diego M Lopez, and Bernd Blobel. Architectural approach for semantic EHR systems development based on Detailed Clinical Models. *Studies in health technology and informatics*, 177:164–9, Jan 2012.
- [97] O Bodenreider. Biomedical ontologies in action: role in knowledge management, data integration and decision support. *Yearb Med Inform*, pages 67–79, Jan 2008.
- [98] Barry Smith, Michael Ashburner, Cornelius Rosse, et al. The OBO Foundry: co-ordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–5, Nov 2007.
- [99] Manuel M Morente, Pedro L Fernández, and Enrique de Alava. Biobanking: old activity or young discipline? *Semin Diagn Pathol*, 25(4):317–22, Nov 2008.
- [100] Peter HJ Riegman, Manuel M Morente, Fay Betsou, et al. Biobanking for better healthcare. *Molecular Oncology*, 2:213–222, Oct 2008.
- [101] Martin Asslaber and Kurt Zatloukal. Biobanks: transnational, European and global networks. *Brief Funct Genomic Proteomic*, 6(3):193–201, Sep 2007.
- [102] Mary E Edgerton, Carl Morrison, Virginia A Livolsi, et al. A Standards based Ontological Approach to Information Handling for use by Organizations Providing Human Tissue for Research. *Cancer informatics*, 6:127–37, Jan 2008.
- [103] D L Ellsworth and T A Manolio. The emerging importance of genetics in epidemiologic research. I. Basic concepts in human genetics and laboratory technology. *Ann Epidemiol*, 9(1):1–16, Jan 1999.
- [104] M M Morente, E de Alava, and P L Fernandez. Tumour banking: the Spanish design. *Pathobiology*, 74(4):245–50, Jan 2007.
- [105] Robert F Ozols, Roy S Herbst, Yolonda L Colson, et al. Clinical cancer advances 2006: major research advances in cancer treatment, prevention, and screening—a report from the American Society of Clinical Oncology. *J Clin Oncol*, 25(1):146–62, Jan 2007.
- [106] Manfred Schmitt, Karin Mengele, Elisabeth Schueren, et al. European Organisation for Research and Treatment of Cancer (EORTC) Pathobiology Group standard operating procedure for the preparation of human tumour tissue extracts

- suitied for the quantitative analysis of tissue-associated biomarkers. *European Journal of Cancer*, 43(5):835–44, Mar 2007.
- [107] I Teodorovic, P Therasse, A Spatz, M Isabelle, and W Oosterhuis. Human tissue research: EORTC recommendations on its practical consequences. *European Journal of Cancer*, 39:2256–2263, Sep 2003.
 - [108] Susanne Vijverberg, Ann K Daly, and Anke Hilse Maitland van der Zee. Conference Scene: Initiatives on future biobanking in pharmacogenomics. *Pharmacogenomics*, 10(7):1135–8, Jul 2009.
 - [109] H E Wichmann. Genetic epidemiology in Germany—from biobanking to genetic statistics. *Methods of Information in Medicine*, 44(4):584–9, Jan 2005.
 - [110] Peter H J Riegman, Antonio Llombart Bosch, and OECI TuBaFrost Consortium. OECI TuBaFrost tumor biobanking. *Tumori*, 94(2):160–3, Jan 2008.
 - [111] Martin Yuille, Gert-Jan van Ommen, Christian Bréchet, et al. Biobanking for Europe. *Briefings in Bioinformatics*, 9(1):14–24, Jan 2008.
 - [112] Bruno Clément, Geneviève Chêne, and Françoise Degos. A national collection of liver tumours: lessons learnt from 6 years of biobanking in France. *Cancer Lett*, 286(1):140–4, Dec 2009.
 - [113] Martin Yuille, Katherine Dixon, Andrew Platt, et al. The UK DNA banking network: a ”fair access” biobank. *Cell Tissue Bank*, Aug 2009.
 - [114] C E Teunissen, A Petzold, J L Bennett, et al. A consensus protocol for the standardization of cerebrospinal fluid collection and biobanking. *Neurology*, 73(22):1914–22, Dec 2009.
 - [115] Maurilio Ponzoni, Ivo Kwee, Luca Mazzucchelli, et al. A virtual tissue bank for primary central nervous system lymphomas in immunocompetent individuals. *Pathobiology*, 74(4):264–9, Jan 2007.
 - [116] Anne Cambon-Thomsen, Pascal Ducournau, Pierre-Antoine Gourraud, and David Pontille. Biobanks for genomics and genomics for biobanks. *Comparative and Functional Genomics*, 4:628–634, Nov 2003.
 - [117] J K Field, T Liloglou, A Niaz, et al. EUELC project: a multi-centre, multipurpose study to investigate early stage NSCLC, and to establish a biobank for ongoing collaboration. *Eur Respir J*, 34(6):1477–86, Dec 2009.

- [118] Leena Peltonen and GenomEUtwin. GenomEUtwin: a strategy to identify genetic influences on health and disease. *Twin Res*, 6(5):354–60, Oct 2003.
- [119] A Cambon-Thomsen, E Rial-Sebbag, and B M Knoppers. Trends in ethical and legal frameworks for the use of human biobanks. *Eur Respir J*, 30(2):373–82, Aug 2007.
- [120] M M Morente, R Mager, S Alonso, et al. TuBaFrost 2: Standardising tissue collection and quality control procedures for a European virtual frozen tissue bank network. *European Journal of Cancer*, 42(16):2684–91, Nov 2006.
- [121] P H J Riegman, W N M Dinjens, and J W Oosterhuis. Biobanking for interdisciplinary clinical research. *Pathobiology*, 74(4):239–44, Jan 2007.
- [122] Rajiv Dhir. Prostate cancer biobanking. *Curr Opin Urol*, 18(3):309–14, May 2008.
- [123] S R Mager, M H A Oomen, M M Morente, et al. Standard operating procedure for the collection of fresh frozen tissue samples. *European Journal of Cancer*, 43(5):828–34, Mar 2007.
- [124] Alex J Rai, Craig A Gelfand, Bruce C Haywood, et al. HUPO Plasma Proteome Project specimen collection and handling: towards the standardization of parameters for plasma proteome samples. *Proteomics*, 5(13):3262–77, Aug 2005.
- [125] M Casparie, A T M G Tiebosch, G Burger, et al. Pathology databanking and biobanking in the Netherlands, a central role for PALGA, the nationwide histopathology and cytopathology data network and archive. *Cellular Oncology*, 29:19–24, Jan 2007.
- [126] Association of Directors of Anatomic and Surgical Pathology. Recommendations for the reporting of pleural mesothelioma. *Am J Clin Pathol*, 127(1):15–9, Jan 2007.
- [127] Ashokkumar A Patel, John R Gilbertson, Anil V Parwani, et al. An informatics model for tissue banks—lessons learned from the Cooperative Prostate Cancer Tissue Resource. *BMC Cancer*, 6:120, Jan 2006.
- [128] John Quackenbush. Standardizing the standards. *Mol Syst Biol*, 2:2006.0010, Jan 2006.
- [129] Andrew R Jones, Angel Pizarro, Paul Spellman, Michael Miller, and FuGE Working Group. FuGE: Functional Genomics Experiment Object Model. *OMICS*, 10(2):179–84, Aug 2006.

- [130] Norman Morrison, Guy Cochrane, Nadeem Faruque, et al. Concept of sample in OMICS technology. *OMICS*, 10(2):127–37, Aug 2006.
- [131] R A Young. Biomedical discovery with DNA arrays. *Cell*, 102(1):9–15, Jul 2000.
- [132] D J Lockhart and E A Winzeler. Genomics, gene expression and DNA arrays. *Nature*, 405(6788):827–36, Jun 2000.
- [133] P O Brown and D Botstein. Exploring the new world of the genome with DNA microarrays. *Nat Genet*, 21(1 Suppl):33–7, Dec 1999.
- [134] Philip Jones, Richard G Côté, Lennart Martens, et al. PRIDE: a public repository of protein and peptide identifications for the proteomics community. *Nucleic Acids Research*, 34(Database issue):D659–63, Jan 2006.
- [135] Lyle D Burgoon. The need for standards, not guidelines, in biological data reporting and sharing. *Nature Biotechnology*, 24(11):1369–73, Nov 2006.
- [136] MGED. A guide to microarray experiments—an open letter to the scientific journals. *Lancet*, 360(9338):1019; author reply 1019, Sep 2002.
- [137] Opinion. Free and public expression. *Nature*, 410(6831):851, Apr 2001.
- [138] Andrew R Jones, Michael Miller, Ruedi Aebersold, et al. The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics. *Nature Biotechnology*, 25(10):1127–33, Oct 2007.
- [139] William Stafford Noble. Data hoarding is harming proteomics. *Nature Biotechnology*, 22(10):1209; discussion 1209, Oct 2004.
- [140] Casper J Albers, Ritsert C Jansen, Jan Kok, Oscar P Kuipers, and Sacha Aft van Hijum. SIMAGE: simulation of DNA-microarray gene expression data. *BMC Bioinformatics*, 7:205, Jan 2006.
- [141] Vattipally B Sreenu, Pankaj Kumar, Javaregowda Nagaraju, and Hampapathalu A Nagarajaram. Microsatellite polymorphism across the *M. tuberculosis* and *M. bovis* genomes: implications on genome evolution and plasticity. *BMC Genomics*, 7:78, Jan 2006.
- [142] John Quackenbush. Data Standards for 'omic' science. *Nature Biotechnology*, 22(5):613–614, Apr 2004.
- [143] Chris F Taylor. Minimum reporting requirements for proteomics: a MIAPE primer. *Proteomics*, 6 Suppl 2:39–44, Sep 2006.

- [144] A Brazma, A Robinson, G Cameron, and M Ashburner. One-stop shop for microarray data. *Nature*, 403(6771):699–700, Feb 2000.
- [145] Spyros Koutoulas and Ronny Siebes. The chipping forecast. Special supplement to *Nature Genetics* Vol 21, 1999.
- [146] Ola Larsson and Rickard Sandberg. Lack of correct data format and comparability limits future integrative microarray research. *Nature Biotechnology*, 24(11):1322–3, Nov 2006.
- [147] Chris F Taylor, Henning Hermjakob, Randall K Julian, et al. The work of the Human Proteome Organisation’s Proteomics Standards Initiative (HUPO PSI). *OMICS*, 10(2):145–51, Aug 2006.
- [148] Oliver Fiehn, Bruce Kristal, Ben van Ommen, et al. Establishing reporting standards for metabolomic and metabonomic studies: a call for participation. *OMICS*, 10(2):158–63, Aug 2006.
- [149] Eric W Deutsch, Catherine A Ball, Jules J Berman, et al. Minimum information specification for in situ hybridization and immunohistochemistry experiments (MISFISHIE). *Nature Biotechnology*, 26(3):305–12, Mar 2008.
- [150] Josef Spidlen, Robert C Gentleman, Perry D Haaland, et al. Data standards for flow cytometry. *OMICS*, 10(2):209–14, Aug 2006.
- [151] Catherine A Ball and Alvis Brazma. MGED standards: work in progress. *OMICS*, 10(2):138–44, Aug 2006.
- [152] Alvis Brazma. Minimum Information About a Microarray Experiment (MIAME) – Successes, Failures, Challenges. *TheScientificWorldJOURNAL*, 9:420–423, Jan 2009.
- [153] Henning Hermjakob. The HUPO proteomics standards initiative—overcoming the fragmentation of proteomics data. *Proteomics*, 6 Suppl 2:34–8, Sep 2006.
- [154] Dawn Field, George Garrity, Tanya Gray, et al. The minimum information about a genome sequence (MIGS) specification. *Nature Biotechnology*, 26(5):541–7, May 2008.
- [155] Susanna-Assunta Sansone, Philippe Rocca-Serra, Weida Tong, et al. A strategy capitalizing on synergies: the Reporting Structure for Biological Investigation (RSBI) working group. *OMICS*, 10(2):164–71, Aug 2006.

- [156] Chris F Taylor. Standards for reporting bioscience data: a forward look. *Drug Discov Today*, 12(13-14):527–33, Jul 2007.
- [157] Chris F Taylor, Dawn Field, Susanna-Assunta Sansone, et al. Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nature Biotechnology*, 26(8):889–96, Aug 2008.
- [158] J Quackenbush. Computational analysis of microarray data. *Nat Rev Genet*, 2(6):418–27, Jun 2001.
- [159] J Aach, W Rindone, and G M Church. Systematic management and analysis of yeast gene expression data. *Genome Res*, 10(4):431–45, Mar 2000.
- [160] Catherine A Ball, Gavin Sherlock, Helen Parkinson, et al. Standards for microarray data. *Science*, 298(5593):539, Oct 2002.
- [161] Christian J Stoeckert and Helen Parkinson. The MGED ontology: a framework for describing functional genomics experiments. *Comparative and Functional Genomics*, 4(1):127–32, Dec 2003.
- [162] Patricia L Whetzel, Helen Parkinson, Helen C Causton, et al. The MGED Ontology: a resource for semantics-based description of microarray experiments. *Bioinformatics*, 22(7):866–73, Mar 2006.
- [163] Opinion. Microarray standards at last. *Nature*, 419(6905):323, Sep 2002.
- [164] Catherine A Ball, Gavin Sherlock, Helen Parkinson, et al. The underlying principles of scientific publication. *Bioinformatics*, 18(11):1409, Oct 2002.
- [165] William B Mattes, Syril D Pettit, Susanna-Assunta Sansone, Pierre R Bushel, and Michael D Waters. Database development in toxicogenomics: issues and efforts. *Environ Health Perspect*, 112(4):495–505, Mar 2004.
- [166] Norman Morrison, A Joseph Wood, David Hancock, et al. Annotation of environmental OMICS data: application to the transcriptomics domain. *OMICS*, 10(2):172–8, Aug 2006.
- [167] Paola Garosi, Carlotta De Filippo, Marjan van Erk, et al. Defining best practice for microarray analyses in nutrigenomic studies. *Br J Nutr*, 93(4):425–32, Apr 2005.
- [168] Tim F Rayner, Philippe Rocca-Serra, Paul T Spellman, et al. A simple spreadsheet-based, MIAME-supportive format for microarray data: MAGE-TAB. *BMC Bioinformatics*, 7:489, Dec 2006.

- [169] Kazuho Ikeo, Jun Ishi-i, Takurou Tamura, Takashi Gojobori, and Yoshio Tateno. CIBEX: center for information biology gene expression database. *C R Biol*, 326(10-11):1079–82, Jan 2003.
- [170] H Parkinson, U Sarkans, M Shojatalab, et al. ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 33(Database issue):D553–5, Jan 2005.
- [171] Alvis Brazma, Helen Parkinson, Ugis Sarkans, et al. ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 31(1):68–71, Jan 2003.
- [172] Ugis Sarkans, Helen Parkinson, Gonzalo Garcia Lara, et al. The ArrayExpress gene expression database: a software engineering and implementation perspective. *Bioinformatics*, 21(8):1495–501, Apr 2005.
- [173] Alvis Brazma, Ugis Sarkans, Alan Robinson, et al. Microarray data representation, annotation and storage. *Adv Biochem Eng Biotechnol*, 77:113–39, Jan 2002.
- [174] Ron Edgar and Tanya Barrett. NCBI GEO standards and services for microarray data. *Nature Biotechnology*, 24(12):1471–2, Dec 2006.
- [175] Alvis Brazma, Helen Parkinson, and EMBL-EBI ArrayExpress team. ArrayExpress service for reviewers/editors of DNA microarray papers. *Nature Biotechnology*, 24(11):1321–2, Nov 2006.
- [176] Catherine A Ball, Ihab A B Awad, Janos Demeter, et al. The Stanford Microarray Database accommodates additional microarray platforms and data formats. *Nucleic Acids Research*, 33(Database issue):D580–2, Dec 2005.
- [177] M P Washburn, D Wolters, and J R Yates. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nature Biotechnology*, 19(3):242–7, Mar 2001.
- [178] Michael Hamacher and Helmut E Meyer. HUPO Brain Proteome Project: aims and needs in proteomics. *Expert review of proteomics*, 2(1):1–3, Jan 2005.
- [179] Gilbert S Omenn, David J States, Marcin Adamski, et al. Overview of the HUPO Plasma Proteome Project: results from the pilot phase with 35 collaborating laboratories and multiple analytical groups, generating a core dataset of 3020 proteins and a publicly-available database. *Proteomics*, 5(13):3226–45, Aug 2005.

- [180] Henning Hermjakob and Rolf Apweiler. The Proteomics Identifications Database (PRIDE) and the ProteomExchange Consortium: making proteomics data accessible. *Expert review of proteomics*, 3(1):1–3, Feb 2006.
- [181] Kevin Garwood, Thomas McLaughlin, Chris Garwood, et al. PEDRo: a database for storing, searching and disseminating experimental proteomics data. *BMC Genomics*, 5(1):68, Sep 2004.
- [182] Chris F Taylor, Norman W Paton, Kevin L Garwood, et al. A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nature Biotechnology*, 21(3):247–54, Mar 2003.
- [183] Patrick G A Pedrioli, Jimmy K Eng, Robert Hubley, et al. A common open representation of mass spectrometry data and its application to proteomics research. *Nature Biotechnology*, 22:1459–1466, Oct 2004.
- [184] Marc R Wilkins, Ron D Appel, Jennifer E Van Eyk, et al. Guidelines for the next 10 years of proteomics. *Proteomics*, 6(1):4–8, Jan 2006.
- [185] Ralph A Bradshaw, Alma L Burlingame, Steven Carr, and Ruedi Aebersold. Reporting protein identification data: the next generation of guidelines. *Molecular & cellular proteomics : MCP*, 5(5):787–8, May 2006.
- [186] Steven Carr, Ruedi Aebersold, Michael Baldwin, et al. The need for guidelines in publication of peptide and protein identification data: Working Group on Publication Guidelines for Peptide and Protein Identification Data. *Molecular & cellular proteomics : MCP*, 3(6):531–3, May 2004.
- [187] Sandra Orchard, Henning Hermjakob, and Rolf Apweiler. The proteomics standards initiative. *Proteomics*, 3(7):1374–6, Jul 2003.
- [188] Robertson Craig, John P Cortens, and Ronald C Beavis. Open source system for analyzing, validating, and storing protein identification data. *J Proteome Res*, 3(6):1234–42, Jan 2004.
- [189] Sam Hanash and Julio E Celis. The Human Proteome Organization: a mission to advance proteome knowledge. *Molecular & cellular proteomics : MCP*, 1(6):413–4, Jun 2002.
- [190] Frank Gibson, Leigh Anderson, Gyorgy Babnigg, et al. Guidelines for reporting the use of gel electrophoresis in proteomics. *Nature Biotechnology*, 26(8):863–4, Aug 2008.

- [191] Paula J Domann, Satoko Akashi, Coral Barbas, et al. Guidelines for reporting the use of capillary electrophoresis in proteomics. *Nature Biotechnology*, 28(7):654–5, Jul 2010.
- [192] Andrew R Jones, Kathleen Carroll, David Knight, et al. Guidelines for reporting the use of column chromatography in proteomics. *Nature Biotechnology*, 28(7):654, Jul 2010.
- [193] Christine Hoogland, Martin O’Gorman, Philippe Bogard, et al. Guidelines for reporting the use of gel image informatics in proteomics. *Nature Biotechnology*, 28(7):655–6, Jul 2010.
- [194] Chris F Taylor, Pierre-Alain Binz, Ruedi Aebersold, et al. Guidelines for reporting the use of mass spectrometry in proteomics. *Nature Biotechnology*, 26(8):860–1, Aug 2008.
- [195] Pierre-Alain Binz, Robert Barkovich, Ronald C Beavis, et al. Guidelines for reporting the use of mass spectrometry informatics in proteomics. *Nature Biotechnology*, 26(8):862, Aug 2008.
- [196] Sandra Orchard, Lukasz Salwinski, Samuel Kerrien, et al. The minimum information required for reporting a molecular interaction experiment (MIMIx). *Nature Biotechnology*, 25(8):894–898, Aug 2007.
- [197] Henning Hermjakob, Luisa Montecchi-Palazzi, Gary Bader, et al. The HUPO PSI’s molecular interaction format—a community standard for the representation of protein interaction data. *Nature Biotechnology*, 22(2):177–83, Feb 2004.
- [198] Sandra Orchard, Bissan Al-Lazikani, Steve Bryant, et al. Minimum information about a bioactive entity (MIABE). *Nat Rev Drug Discov*, 10(9):661–9, Sep 2011.
- [199] Samuel Kerrien, Sandra Orchard, Luisa Montecchi-Palazzi, et al. Broadening the horizon—level 2.5 of the HUPO-PSI format for molecular interactions. *BMC Biol*, 5:44, Jan 2007.
- [200] Sandra Orchard, Henning Hermjakob, Chris F Taylor, et al. Further steps in standardisation. Report of the second annual Proteomics Standards Initiative Spring Workshop (Siena, Italy 17-20th April 2005). *Proteomics*, 5(14):3552–5, Sep 2005.
- [201] Frank Gibson, Christine Hoogland, Salvador Martinez-Bartolomé, et al. The gel electrophoresis markup language (GelML) from the Proteomics Standards Initiative. *Proteomics*, 10(17):3073–81, Sep 2010.

- [202] HUPO PSI. mzIdentML: exchange format for peptides and proteins identified from mass spectra. Available online at <http://www.psdev.info/files/mzIdentML1.0.0.pdf> [last accessed July 30, 2012].
- [203] Andrew R Jones and Frank Gibson. An update on data standards for gel electrophoresis. *Proteomics*, 7 Suppl 1:35–40, Sep 2007.
- [204] Opinion. Democratizing proteomics data. *Nature Biotechnology*, 25(3):262, Mar 2007.
- [205] David Croft, Gavin O’Kelly, Guanming Wu, et al. Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Research*, 39(Database issue):D691–7, Jan 2011.
- [206] Julie Bourbeillon, Sandra Orchard, Itai Benhar, et al. Minimum information about a protein affinity reagent (MIAPAR). *Nature Biotechnology*, 28(7):650–3, Jul 2010.
- [207] Jamie A Lee, Josef Spidlen, Keith Boyce, et al. MIFlowCyt: the minimum information about a Flow Cytometry Experiment. *Cytometry A*, 73(10):926–30, Oct 2008.
- [208] Dawn Field, George Garrity, Norman Morrison, et al. eGenomics: Cataloguing our Complete Genome Collection. *Comparative and Functional Genomics*, 6(7-8):363–8, Jan 2005.
- [209] Guy Cochrane, Philippe Aldebert, Nicola Althorpe, et al. EMBL Nucleotide Sequence Database: developments in 2005. *Nucleic Acids Research*, 34(Database issue):D10–5, Dec 2006.
- [210] Dawn Field and Jennifer Hughes. Cataloguing our current genome collection. *Microbiology (Reading, Engl)*, 151(Pt 4):1016–9, Mar 2005.
- [211] Dawn Field, George Garrity, Tanya Gray, et al. eGenomics: Cataloguing Our Complete Genome Collection III. *Comparative and Functional Genomics*, 2007, 2007.
- [212] Dawn Field, Linda Amaral-Zettler, Guy Cochrane, et al. The Genomic Standards Consortium. *PLoS Biol*, 9(6):e1001088, Jun 2011.
- [213] Dawn Field, Frank Oliver Glöckner, George M Garrity, et al. Meeting report: the fourth Genomic Standards Consortium (GSC) workshop. *OMICS*, 12(2):101–8, Jun 2008.

- [214] Ioanna Pagani, Konstantinos Liolios, Jakob Jansson, et al. The Genomes OnLine Database (GOLD) v.4: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, 40(Database issue):D571–9, Jan 2012.
- [215] Jie Huang, Daniel Mirel, Elizabeth Pugh, et al. Minimum Information about a Genotyping Experiment (MIGEN). *Stand Genomic Sci*, 5(2):224–9, Nov 2011.
- [216] Eric W Deutsch, Catherine A Ball, G Steven Bova, et al. Development of the Minimum Information Specification for In Situ Hybridization and Immunohistochemistry Experiments (MISFISHIE). *OMICS*, 10(2):205–8, Aug 2006.
- [217] Dawn Field, Susanna-Assunta Sansone, Amanda Collis, et al. Megascience. 'Omics data sharing. *Science*, 326(5950):234–6, Oct 2009.
- [218] Douglas G Altman, Iveta Simera, John Hoey, David Moher, and Ken Schulz. EQUATOR: reporting guidelines for health research. *Open Med*, 2(2):e49–50, Jan 2008.
- [219] Patricia L Whetzel, Ryan R Brinkman, Helen C Causton, et al. Development of FuGO: an ontology for functional genomics investigations. *OMICS*, 10(2):199–204, Aug 2006.
- [220] M Ashburner, C A Ball, J A Blake, et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1):25–9, May 2000.
- [221] Barry Smith, Werner Ceusters, Bert Klagges, et al. Relations in biomedical ontologies. *Genome Biol*, 6(5):R46, Jan 2005.
- [222] Georgios V Gkoutos, Eain C J Green, Ann-Marie Mallon, John M Hancock, and Duncan Davidson. Using ontologies to describe mouse phenotypes. *Genome Biol*, 6(1):R8, Jan 2005.
- [223] Functional Genomics Experiment (FuGE) Community. FuGE: Functional Genomics Experiment model specification. Available online at <http://fuge.sourceforge.net/dev/V1Final/FuGE-v1-SpecDoc.doc> [last accessed October 25, 2012].
- [224] Adam Rauch, Matthew Bellew, Jimmy Eng, et al. Computational Proteomics Analysis System (CPAS): an extensible, open-source analytic system for evaluating and publishing proteomic data and high throughput biological experiments. *J Proteome Res*, 5(1):112–21, Jan 2006.

- [225] Sandra Orchard, Rolf Apweiler, Robert Barkovich, et al. Proteomics and Beyond: a report on the 3rd Annual Spring Workshop of the HUPO-PSI 21-23 April 2006, San Francisco, CA, USA. *Proteomics*, 6(16):4439–43, Aug 2006.
- [226] Morris A Swertz, K Joeri van der Velde, Bruno M Tesson, et al. XGAP: a uniform and extensible data model and software platform for genotype and phenotype experiments. *Genome Biol*, 11(3):R27, Jan 2010.
- [227] Yu Qian, Olga Tchuvatkina, Josef Spidlen, et al. FuGEFlow: data model and markup language for flow cytometry. *BMC Bioinformatics*, 10:184, Jan 2009.
- [228] Susanna-Assunta Sansone, Philippe Rocca-Serra, Marco Brandizi, et al. The first RSBI (ISA-TAB) workshop: "can a simple format work for complex studies?". *OMICS*, 12(2):143–9, Jun 2008.
- [229] Daniel L Rubin, Suzanna E Lewis, Chris J Mungall, et al. National Center for Biomedical Ontology: advancing biomedicine through structured organization of scientific knowledge. *OMICS*, 10(2):185–98, Aug 2006.
- [230] Alejandra N Gonzalez-Beltran, May Y Yong, Gairin Dancey, and Richard H Begent. Guidelines for Information About Therapy Experiments: a proposal on best practice for recording experimental data on cancer therapy. *BMC research notes*, 5(1):10, Jan 2012.
- [231] John Quackenbush, Christian Stoeckert, Catherine Ball, et al. Top-down standards will not serve systems biology. *Nature*, 440(7080):24, Mar 2006.
- [232] Melanie Bettina Späth and Jane Grimson. Applying the archetype approach to the database of a biobank information management system. *I. J. Medical Informatics*, 80(3):205–226, 2011.
- [233] Genetic Engineering and Biotechnology News. Genetix Acquires SlidePath for \$4.89M to Pad Digital Pathology Capabilities. <http://www.genengnews.com/gen-news-highlights/genetix-acquires-slidepath-for-4-89m-to-pad-digital-pathology-capabilities/53687096/> [last accessed August, 2012].
- [234] Leica Microsystems. Advanced Digital Pathology Information Management Distiller. <http://www.leica-microsystems.com/products/digital-pathology/manage/details/product/distiller/> [last accessed August, 2012].
- [235] Sebastian Garde, Evelyn Hovenga, Jana Granz, Shahla Foozonkhah, and Sam Heard. Managing Archetypes for Sustainable and Semantically Interoperable Electronic Health Records. *eJHI electronic Journal of Health Informatics*, 2(2):10, 2007.

- [236] Heather Leslie. Archetype Authoring Process and Lifecycle. <http://www.openehr.org/wiki/display/healthmod/Archetype+authoring%2C+review+and+publication> [last accessed August, 2012].
- [237] *openEHR*. Clinical Models Project. Available online at <http://www.openehr.org/clinicalmodels/project.html> [last accessed August 30, 2012].
- [238] Thomas Beale. The *openEHR* Reference Model: Integration Information Model, Mar 2006. Available online at http://www.openehr.org/releases/1.0.2/architecture/rm/integration_im.pdf [last accessed October 28, 2012].
- [239] HUPO PSI Mass Spectrometry Standards Working Group. mzML: Mass Spectrometry Markup Language. Available online at http://psidev.cvs.sourceforge.net/*checkout*/psidev/psi/psi-ms/mzML/document/mzML1.1.0_specificationDocument.doc [last accessed August 30, 2012].
- [240] Thomas Beale, Sam Heard, Dipak Kalra, and David Lloyd. The *openEHR* Reference Model: Data Structures Information Model, Nov 2008. Available online at http://www.openehr.org/releases/1.0.2/architecture/rm/data_structures_im.pdf [last accessed August, 2012].
- [241] Thomas Beale, Sam Heard, Dipak Kalra, and David Lloyd. The *openEHR* Reference Model: *openEHR* Common Information Model, Dec 2008. Available online at http://www.openehr.org/releases/1.0.2/architecture/rm/common_im.pdf [last accessed October 28, 2012].
- [242] P. H. O’Farrell. High-resolution two-dimensional electrophoresis of proteins. *Journal of Biological Chemistry*, 250:4007–4021, 1975.
- [243] NHS Connecting for Health. Investigating implementing CEN 13606 with HL7 V3 and SNOMED CT – Final Report, Jan 2006. Available online at http://detailedclinicalmodels.org/documents/NHS_CFH_13606InvestigationRpt_v1-0.pdf [last accessed July 30, 2012].
- [244] D. Kalra, A. Tapuria, T. Austin, and G. De Moor. Quality requirements for EHR Archetypes. *Stud Health Technol Inform*, 180, 2012.
- [245] Jesualdo Tomas Fernandez-Breis, Marcos Menarguez-Tortosa, Catalina Martinez-Costa, et al. A Semantic Web-based System for Managing Clinical Archetypes. *Conference proceedings : 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Conference*, pages 1482–1485, Sep 2008.

- [246] Ronald Cornet. Definitions and qualifiers in SNOMED CT. *Methods Inf Med*, 48(2):178–83, Jan 2009.
- [247] Robert H Dolin, Kent A Spackman, and David Markwell. Selective retrieval of pre- and post-coordinated SNOMED concepts. *Proc AMIA Symp*, pages 210–4, Jan 2002.
- [248] S Trent Rosenbloom, Randolph A Miller, Kevin B Johnson, Peter L Elkin, and Steven H Brown. Interface terminologies: facilitating direct entry of clinical data into electronic health record systems. *J Am Med Inform Assoc*, 13(3):277–88, Jan 2006.
- [249] Dan Dalan. Clinical data mining and research in the allergy office. *Curr Opin Allergy Clin Immunol*, 10(3):171–7, Jun 2010.
- [250] Sookyung Hyun, Stephen B Johnson, and Suzanne Bakken. Exploring the ability of natural language processing to extract data from nursing narratives. *Comput Inform Nurs*, 27(4):215–23; quiz 224–5, Jan 2009.
- [251] Stephen B Johnson, Suzanne Bakken, Daniel Dine, et al. An electronic health record based on structured narrative. *J Am Med Inform Assoc*, 15(1):54–64, Jan 2008.
- [252] Krister J Kristianson, Henrik Ljunggren, and Lars L Gustafsson. Data extraction from a semi-structured electronic medical record system for outpatients: a model to facilitate the access and use of data for quality control and research. *Health Informatics J*, 15(4):305–19, Dec 2009.
- [253] S M Meystre, G K Savova, K C Kipper-Schuler, and J F Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform*, pages 128–44, Jan 2008.
- [254] Guergana K Savova, James J Masanz, Philip V Ogren, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc*, 17(5):507–13, Sep 2010.
- [255] Irena Spasic, Farzaneh Sarafraz, John A Keane, and Goran Nenadic. Medication information extraction with linguistic pattern matching and semantic rules. *J Am Med Inform Assoc*, 17(5):532–5, Sep 2010.
- [256] Jinsoo Park and Sudha Ram. Information Systems Interoperability: What Lies Beneath? *ACM Transactions on Information Systems*, 22(4):1595–632, Sep 2010.

- [257] Rong Chen. *openEHR* Reference Model Java ITS, Feb 2006. Available online at <http://www.openehr.org/releases/1.0.1/its/Java/openEHR-JavaITS.pdf> [last accessed July 30, 2012].

Appendix A

Details of the PCRC BIMS database schema

Table A.1 gives detailed explanations of the fields in the PCRC BIMS database. New tables are marked with yellow background colour. Fields for which no further explanation was provided are considered self-explanatory and are labelled “OK”. SOP refers to Standard Operating Procedure.

PCRC BIMS table fields	Explanation
Patient	OK
PCRC Study Number (Number)	At each hospital a patient is initially given a unique anonymous ID code. For example at SJH this could be IMM0001. Once the details of this patient (i.e. patient IMM0001) are entered into the system by the research nurse the patient is automatically assigned a unique PCRC number by the BIMS, which is the number seen by researchers.
Institute Study Number (short text)	To identify a sample the 'Institute Study Number' is used. The site of collection + a unique number allocated for that patient is printed on a label and this is attached to each individual sample (aliquot) for that patient. For Example, the codes at a particular site could be as follows: Mater Hospital: MMUH09/001 Beaumont Hospital: BH09/001 St James's Hospital: IMM001 (IMM stands for INstitute of Molecular Medicine, which is the TCD research centre based at St James's Hospital)
Race (choice: Asian, Caucasian,...)	OK
Current Medication (long text)	The 'Current Medications Field' includes ALL medications that the patient is taking. This can include medications for other ongoing medical issues aside from prostate cancer. On occasion a patient may have recieved hormone therapy prior to a decision to go for surgery and the therapy they were on for this will be recorded in the medications section. So essentially any medications they are on when the medical history is taken by a urologist/admitting nurse/research nurse consenting the patient. Information on therapies/medications given AFTER surgery will be input in the treatment options table.
Current Medical History (long text)	What medical issues the patient currently has which have been diagnosed by a doctor that they may or may not be on medication for, are current. For example, if they have hypertension now and may be on medication for it, this is entered into Current Medical History. Currently at SJH the Urology Registrar that consents in patient will record this information on a proforma and it is then entered in the BIMS. At Beaumont Hospital and The Mater the research nurse records this information and enters it into the BIMS.

Past Medical/Surgical History (long text)	Any medical conditions the patient has had in the past but does not currently suffer from and is not on any medication for, are past medical issues. For example, if they had an appendectomy 10 years ago this is in Past medical/surgical history. Currently at SJH the Urology Registrar that consents in patient will record this information on a proforma and it is then entered in the BIMS. At Beaumont Hospital and The Mater the research nurse records this information and enters it into the BIMS.
Current Symptoms (short text)	OK
Family History (long text)	OK
BPH	BPH stands for Benign Prostatic Hyperplasia. A TURP, i.e. a transurethral resection of the prostate is performed to treat BPH. Urine and Blood samples are collected for these patients. When details for the samples for TURPS are entered the collected at button is selected as being TURP. Tissue chips taken at TURP are examined by a pathologist and retained by each hospitals pathology department. If cancer is present (indicated by Gleason Score) this is recorded in the BPH table. Fresh frozen tissue is not collected at TURP.
Procedure ID (Number)	For use within the database. Data is not entered into these fields.
Date of Surgery (Date)	OK
Age at Time of Surgery (Number)	OK
Gleason Grade A (Number)	Gleason Grades (and scores) can be reported for % of an individual biopsy core or % of the entire biopsy tissue. There can be more than one Gleason Score reported and documented in the biopsy report by the pathologist. Currently the highest Gleason Grade reported is entered in the BIMS. For example if 3+3=6 is reported as being present in one core and 4+3=7 is reported as being present in a another core, 4+3=7 is input.
Gleason Grade B (Number)	See explanation for Gleason Grade A
Gleason Score (A + B) (formula: A+B)	In future: will use tertiary GS as well.
Gleason Score (Legacy) (Number)	Very rare, sometimes in biopsy, not occurred in radical prost. This field was included in the event that individual Gleason Grades (A or B) are not given in the pathology report and only an overall Gleason Score is reported. The value can be 4-10. Never come across a case when we needed to use this field.
Biopsy	refers to Fine Needle Aspiration Biopsy (FNAB)
Biopsy Number (Number)	For use within the database. Data is not entered into these fields.
Date of Biopsy (Date)	OK
Hospital (choice: Beaumont, St. James, St. Vincents,...)	OK
Number of Cores (Number)	Total number of cores taken during prostate biopsy. Cores: have sizes but not used in BIMS yet. Cores not consented for experiments, hope for future use. Cores needed to be annotated individually, but are not done as routine. Overall highest Gleason regardless of volume will be given for biopsy.
No. Cores - Peripheral left (PL) (Number)	Peripheral refers to peripheral zone in prostate: http://en.wikipedia.org/wiki/Prostate
PL Number of Cores Involved (Number)	Peripheral refers to peripheral zone in prostate: http://en.wikipedia.org/wiki/Prostate
PL % Core Involvement (Number)	Peripheral refers to peripheral zone in prostate: http://en.wikipedia.org/wiki/Prostate
No. Cores - Transition left (TL) (Number)	Transition refers to transition zone in prostate: http://en.wikipedia.org/wiki/Prostate
TL Number of Cores Involved (Number)	Transition refers to transition zone in prostate: http://en.wikipedia.org/wiki/Prostate
TL % Core Involvement (Number)	Transition refers to transition zone in prostate: http://en.wikipedia.org/wiki/Prostate
No. Cores - Peripheral Right (PR) (Number)	Peripheral refers to peripheral zone in prostate: http://en.wikipedia.org/wiki/Prostate

PR Number of Cores Involved (Number)	Peripheral refers to peripheral zone in prostate: http://en.wikipedia.org/wiki/Prostate
PR % Core Involvement (Number)	Peripheral refers to peripheral zone in prostate: http://en.wikipedia.org/wiki/Prostate
No. Cores - Transition Right (TR) (Number)	Transition refers to transition zone in prostate: http://en.wikipedia.org/wiki/Prostate
TR Number of Cores Involved (Number)	Transition refers to transition zone in prostate: http://en.wikipedia.org/wiki/Prostate
TR % Core Involvement (Number)	Transition refers to transition zone in prostate: http://en.wikipedia.org/wiki/Prostate
Gleason Grade - A (Number)	as before
Gleason Grade - B (Number)	as before
Gleason Grade (Legacy) (Number)	as before
Gleason Score (A + B) (Formula: A+B)	as before
cTNM Staging - T (choice: 2, 4a, 3c, 3b, 3a, 1a, 1b, 1c, 2b, 2a, 2c, 4b)	clinical TNM staging
cTNM Staging - N (choice: 0,1,2,3)	clinical TNM staging
cTNM Staging - M (choice: 0,1)	clinical TNM staging
DRE (choice: Negative, Positive)	examination if positive, describe what is being felt, reg does
Radical Prostatectomy	OK
Procedure ID (Number)	For use within the database. Data is not entered into these fields.
Date of Surgery (Date)	OK
Age at Time of Surgery (Number)	OK
Gleason Grade A (Number)	as before
Gleason Grade B (Number)	as before
Gleason Score (A + B) (formula: A+B)	as before
Gleason Score (Legacy) (Number)	as before
pTNM Staging - T (choice: 2, 4a, 3c, 3b, 3a, 1a, 1b, 1c, 2b, 2a, 2c, 4b)	pathological TNM staging
pTNM Staging - N (choice: 0,1,2,3)	pathological TNM staging
pTNM Staging - M (choice: 0,1)	pathological TNM staging
Capsular Invasion? (choice: No/Yes)	invasion of cancer into the prostatic capsule
Capsular Invasion: Extra Capsular Extension?(choice: No/Yes)	http://www.phoenix5.org/glossary/extracapsular_extension.html , Extracapsular extension is the spreading of prostate cancer out of the capsule of the prostate cancer also known as the prostatic capsule.
Extra Capsular Extension: Posterior? (choice: No/Yes)	See explanation for Capsular Invasion
Extra Capsular Extension: Left-/Right/Both? (choice: Left-/Right/Both)	See explanation for Capsular Invasion
Extra Capsular Extension: Apical? (choice: No/Yes)	See explanation for Capsular Invasion

Margins? (choice: No/Yes)	Positive Margins means that there are cancer cells at the edge of tissue that has been removed. This can only be seen under the microscope, after the surgery. Positive margins occur about 25-35% of the time after non-nerve sparing radical prostatectomies and 35-45% of the time in nerve sparing radical prostatectomies. Positive margins mean the chance of the cancer recurring in the surgical area is significantly greater than if the margins are negative. Margins: It is the goal of surgery to remove all the cancer. A positive margin means that the pathologist notices that cancer cells are at the very edge of the prostate touching the ink that was applied during initial processing of the prostate gland. The pathologist will note the number and location of any positive margins. Those patients with positive surgical margins are at an increased risk of cancer recurrence. Patients with more than one positive margin are more likely to have cancer recur compared to those with a single positive margin. Patients with an extensive positive margin (large area where the cancer is in contact with the edge of the prostate) are more likely to have recurrence of their cancer compared to those with a very small area (focal positive margins) where the cancer just touches the edge. - Margin fields may be reformatted in future edition
Margins: Posterior? (choice: No/Yes)	See explanation for Margins - margin fields may be reformatted in future edition
Margins: Left/Right/Both? (choice: Left/Right/Both)	See explanation for Margins - margin fields may be reformatted in future edition
Margins: Apical? (choice: No/Yes)	See explanation for Margins - margin fields may be reformatted in future edition
PIN (choice: Absent/Present)	Prostatic Intraepithelial Neoplasia is believed to be the most likely precursor of prostate cancer. It is a proliferative lesion i.e. it is composed of prostatic epithelial cells that are dividing more rapidly than normal epithelium. However, the cells have not yet become cancerous. Pin can be classified as high grade, medium grade and low grade. Low grade PIN is less worrisome than the other two grades.
PIN Grade (choice: Low/High)	See explanation for PIN
Focal Atrophy (choice: No/Yes)	http://demarzolab.pathology.jhmi.edu/AtrophyClassification/StaticPage.html
Atypia (No/Yes)	Studies also point to a condition known as Atypical Small Acinar Proliferation, known as Atypia or ASAP, as another cause of cancer. Under a microscope the cells look cancerous although their numbers are too small to be certain. There is a 40% to 50% chance of prostate cancer when ASAP is present. Doctors will also direct the patient to obtain a repeat biopsy after a few months.
Lymph node involvement (choice: Negative/Positive)	Cancer has spread to lymph nodes
Seminal Vesicles Involvement (choice: Negative/Positive)	cancer has spread to seminal vesicles. Seminal vesicles are connected to the vas deferens which transport sperm to them before ejaculation. Seminal vesicles make up the bulk of the ejaculatory fluid. After adding fluid to the sperm, the liquid is transported to the prostate gland. The prostate gland adds hormones and proteins which nourish the sperm and keep them alive long enough to reach an egg. The prostate gland also adds an anti-coagulant which helps keep the ejaculatory fluid from congealing. Removal of the seminal vesicles during prostate cancer treatment, along with the prostate gland in radical prostatectomy, can cause dry orgasm. Men experience the same sensations associated with a normal orgasm, but do not produce ejaculatory fluid.
Perineural Involvement (choice: Negative/Positive)	cancer has spread to perineum (http://en.wikipedia.org/wiki/Perineum)
Volume of Prostate (Number)	OK
% gland involved (Number)	OK
Involvement of Inflammatory Cells (choice: No/Yes/Not Reported)	Maybe Deleting the field not used. Inflammation found only rarely.
Biochemistry	OK

Biochemistry ID (Number)	For use within the database. Data is not entered into these fields.
Biochemistry Laboratory (short text)	OK
Pre-Op PSA Velocity (Number)	not clear what formula and which PSA values are used for it
Post-Op PSA Velocity (Number)	not clear what formula and which PSA values are used for it
Pre-Op PSA	any PSA measurement (http://en.wikipedia.org/wiki/Prostate-specific_antigen) BEFORE radical prostatectomy
Test ID (Number)	For use within the database. Data is not entered into these fields.
Date (Date)	Date of sample taken, in chart if only month/year then use 1st of month/year
PSA (Number)	PSA test measures the level of PSA (http://en.wikipedia.org/wiki/Prostate-specific_antigen) in the blood. A blood sample can be taken by a GP and then sent for testing to the hospital, or a blood sample can be taken at the hospital and tested there.test carried out by medical laboratory scientists. They are recorded in the patient chart or on the hospitals computer system and would have been part of arriving at a diagnosis of prostate cancer. This blood is not kept as a sample by the Bio-resource. Not known if the sample is discarded or stored by the diagnostic laboratory once the PSA reading is obtained.
Free/Total PSA (Number)	Most PSA in the blood is bound to serum proteins. A small amount is not protein bound and is called free PSA. In men with prostate cancer the ratio of free (unbound) PSA to total PSA is decreased. The risk of cancer increases if the free to total ratio is less than 25%. The lower the ratio the greater the probability of prostate cancer. Measuring the ratio of free to total PSA appears to be particularly promising for eliminating unnecessary biopsies in men with PSA levels between 4 and 10 ng/mL.[19] However, both total and free PSA increase immediately after ejaculation, returning slowly to baseline levels within 24 hours.[12]
Post-Op PSA	any PSA measurement (http://en.wikipedia.org/wiki/Prostate-specific_antigen) AFTER radical prostatectomy. Follow up PSAs are collected for each patient for 5 years after their radical prostatectomy surgery.
Test ID (Number)	For use within the database. Data is not entered into these fields.
Date (Date)	Date of sample taken, in chart if only month/year then use 1st of month/year
PSA (Number)	as before
Free/Total PSA (Number)	as before
Patient Outcome	
Report Number (Number)	For use within the database. Data is not entered into these fields.
Post-Op Rising PSA (Number)	If the PSA rises, this PSA value which indicates a biochemical recurrence is input here. high post op-psa before further treatment, upper level is 4. A rise in PSA to 1 or >1 would signal that a possible recurrence has occurred. At this time the patient will be closely monitored and the frequency of PSA testing will increase to once every 3 months. If a recurrence has occurred this may be local (for example in the prostate bed, lymph nodes) or more widespread to the bones (metastatic/systemic).
Bone Pain (choice: No/Yes)	OK
Diagnosis (long text)	Information relating to the diagnosis of a recurrence of prostate cancer, which could include description of physical symptoms including bone pain, also perhaps a rise in PSA etc.
Bone Scan (short text)x	description of scan results. http://prostatecancer.about.com/od/symptoms_anddiagnosis/a/bonescanbasics.htm
Diagnosis - Local or Systemic (choice: Local/Systemic)	whether the cancer is a local recurrence or metastatic based on the results of the bone scan.
Treatment Options	see http://www.drcatalona.com/quest/quest.spring03.3.htm for overview
Treatment ID (Number)	For use within the database. Data is not entered into these fields.
Hormone Therapy? (choice: No/Yes)	http://www.prostatehealth.org.au/philp.5.html , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.

Hormone Therapy: Type of therapy (long text)	http://www.prostatehealth.org.au/philp_5.html , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Radiotherapy? (choice: No/Yes)	http://www.virtualcancercentre.com/treatments.asp?sid=12 , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Radiotherapy: Type of therapy (long text)	http://www.virtualcancercentre.com/treatments.asp?sid=12 , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Radiotherapy: Hospital (choice: Mater, St. James, St. Vincents)	http://www.virtualcancercentre.com/treatments.asp?sid=12 , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Radiotherapy: Dose (Gy) (Number)	http://www.virtualcancercentre.com/treatments.asp?sid=12 , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Chemotherapy? (choice: No/Yes)	Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Chemotherapy: Type of therapy (choice: short text – should be long?)	http://www.upmccancercenters.com/cancer/prostate/chemotherapy.html , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Chemotherapy: Hospital (choice: Mater, St. James, St. Vincents)	Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Chemotherapy: Drug (short text)	http://www.upmccancercenters.com/cancer/prostate/chemotherapy.html , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Chemotherapy: Dose (Number)	http://www.upmccancercenters.com/cancer/prostate/chemotherapy.html , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Chemotherapy: Length of treatment (Number)	http://www.upmccancercenters.com/cancer/prostate/chemotherapy.html , Information on therapies/medications given AFTER surgery will be input in the treatment options table. history of treatment.
Outcome	OK
Outcome ID (Number)	For use within the database. Data is not entered into these fields.
Time to Hormone Refractory disease (Months) (Number)	starting time probably from radical prostatectomy to be confirmed. When referring to prostate cancer, disease that is no longer sensitive to the beneficial effects of hormone treatments.
Time to Biochemical Recurrence (Months) (Number)	starting time probably from radical prostatectomy to be confirmedBiochemical recurrence = rise in blood level of PSA in prostate cancer patients after treatment with surgery or radiation = interpretation of the biochemistry result/PSA velocity
Time to Clinical Recurrence (Months) (Number)	starting time probably from radical prostatectomy to be confirmedProstate cancer is back after initial treatment
Date of Death (Date)	OK
Cause (Short text)	OK
Sample Collection – Blood	see SOP: “Collection of Whole Blood Sample for RNA Isolation and Analysis” and SOP: “RNA extraction from whole blood Tempus Spin Kit”
Blood Sample Identifier (short text)	For use within the database. Data is not entered into these fields.
Date Collected (Date)	OK
Time of Collection (Time)	OK
Time of Spinning (Time)	SOP: “Collection of Whole Blood Sample for RNA Isolation and Analysis” and SOP: “RNA extraction from whole blood Tempus Spin Kit”
Time to Process (Time)	From time of collection to time of spinning.
Plasma	Blood processing steps to get Plasma: Blood is spun, frozen
Location ID (Number)	For use within the database. Data is not entered into these fields.
Number of Tubes (Number)	handled as unit all named the same. System updated whenever tubes removed

Location - Freezer Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Location - Box Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
PLASMA Status (choice: Usable/Unusable)	Indicates usability status for proteomics experiment
Serum	Blood processing steps to get Serum: Blood is spun, frozen
Location ID (Number)	For use within the database. Data is not entered into these fields.
Number of Tubes (Number)	handled as unit all named the same. System updated whenever tubes removed
Location - Freezer Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Location - Box Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
SERUM Status (choice: Usable/Unusable)	Indicates usability status for proteomics experiment
DNA	Blood processing steps to get DNA: Blood is spun, frozen, then commercial DNA automated extraction system PureGene Chemistry
Location ID (Number)	For use within the database. Data is not entered into these fields.
Number of Tubes (Number)	handled as unit all named the same. System updated whenever tubes removed
Location - Freezer Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Location - Box Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
RNA	Blood processing steps to get RNA: Blood is spun, frozen until RNA collected in Tempus tube with stabilization liquid, see SOP: "Collection of Whole Blood Sample for RNA Isolation and Analysis" and SOP: "RNA extraction from whole blood Tempus Spin Kit"
Location ID (Number)	For use within the database. Data is not entered into these fields.
Number of Tubes (Number)	handled as unit all named the same. System updated whenever tubes removed
Location - Freezer Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Location - Box Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Fresh - RNA Later	This refers to fresh frozen tissue collected at radical prostatectomy only. When entering information on the tissue there is an option to select 'collected at'. At this time this will always be radical prostatectomy. Prostate sample: slides and blocks: e.g. 50 slides a1-a50. Pathologist will mark out Gleason score on slides not as routine but if asked. Histologically normal/cancer/high grade PIN/BBH, but could be more. Formalin fixed paraffin embedded tissue, sample dealt with in batch
Location ID (Number)	For use within the database. Data is not entered into these fields.
Sample Type (choice: Normal/Tumour)	Pathologist's decision
Location - Freezer Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Location - Box Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Sample Collection – Tissue	This refers to fresh frozen tissue collected at radical prostatectomy only. When entering information on the tissue there is an option to select 'collected at'. At this time this will always be radical prostatectomy. Prostate sample: slides and blocks: e.g. 50 slides a1-a50. Pathologist will mark out Gleason score on slides not as routine but if asked. Histologically normal/cancer/high grade PIN/BBH, but could be more. Formalin fixed paraffin embedded tissue, see SOP 2: Collection of Prostate and Dissection of Samples.
Diagnostic Sample Pathology Number (short text)	For use within the database. Data is not entered into these fields.

Date Collected (Date)	time of radical prostatectomy operation, since its the only time that tissue is collected that is stored in PCRC
Sample Collection – Urine	SOP missing!
Urine Sample Identifier (short text)	For use within the database. Data is not entered into these fields.
Date Collected (Date)	OK
Time of Voiding (Time)	time of voiding = time of collection
Time of Processing (Time)	At SJH (St Jamess Hospital) this is currently recorded as time of addition of protease inhibitor tablet protease: inhibitor tablet is put in the jar of urine collection, for genomic. Protease inhibitor happens after the dip different in different hospitals, some before, some after. Varies between hospitals: Time of processing is urinalysis test
Time to Process (Time)	=Time of Processing minus Time of Voiding: At SJH (St Jamess Hospital) this is currently recorded as Time of Processing (i.e. start of the processing of the urine once collected from the hospital, which is the time of addition of protease inhibitor tablet) minus the Time of Voiding.
Urine pH (Number)	http://en.wikipedia.org/wiki/Urinalysis
Urinalysis - Protein (choice: No/Yes)	http://en.wikipedia.org/wiki/Urinalysis
Urinalysis - Blood (choice: No/Yes)	http://en.wikipedia.org/wiki/Urinalysis
Location - Freezer Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Location - Box Number (short text)	Shelf, rack in shelf, space, box number 1.1.1
Preoperative Fluid? (choice: No/Yes)	Pre-operative fluids are IV fluids given to the patient prior to their surgery. Pre-operative fluids are generally not given at SJH. Sometimes given in BH (Beaumont Hospital) and the Mater pre operatively depending on how long the patient is fasting.
Amount of Preop Fluid (Number)	See explanation for Preoperative Fluid field

Table A.1: Explanation of the fields in the tables of the PCRC BIMS database

Appendix B

The MIAPE-MS guidelines and checklist

The MIAPE-MS guidelines

Reporting guidelines for mass spectrometry

1. General features

1.1 Global descriptors

- Responsible person (or institutional role if more appropriate); provide name, affiliation and stable contact information
- Instrument manufacturer and model
- Customisations (summary)

2. Ion sources

As each spectrum is acquired using only one ionisation source, select the one that applies

2.1 Electrospray Ionisation (ESI)

- Supply type (static, or fed)
- Interface manufacturer, model
- Sprayer type, manufacturer, model
- Other parameters if discriminant for the experiment

2.2 MALDI

- Plate composition (or type)
- Matrix composition
- PSD (or LID/ISD) summary, if performed
- Laser type and wavelength,

Other laser and source related parameters, if discriminating for the experiment

2.3 Other ionisation source

- Description of the ion source and relevant parameters

3. Post-source component

As an MS spectrum or chromatogram performed on one instrument cannot be acquired using all existing analysers and detectors, select the elements that apply

3.1 Analyser

- Ion optics, simple quadrupole, hexapole, Paul trap, linear trap, magnetic sector, FT-ICR, Orbitrap: name of the analyser(s)
- Time-of-flight drift tube (TOF): Reflectron status

3.2 Activation / dissociation

The associated acquisition parameters are covered in 4.1

- Instrument component where the activation / dissociation occurs
- Gas type (when used)

```

- Activation / dissociation type
4. Spectrum and peak list generation and annotation
4.1 Data acquisition
- Software name and version
- Acquisition parameters

4.2 Data analysis
- Software name and version
- Parameters used in the generation of peak lists or processed spectra

4.3 Resulting data
The following information should be provided for each dataset
- Location of source (raw?) and processed files
- The chromatogram(s) for SRM data and other relevant cases

The following information should be provided for each spectrum or peaklist
- m/z and intensity values
- MS level
- Ion mode
- For MS level 2 and higher, precursor m/z and charge if known, with the full mass
  spectrum / peaklist containing that precursor peak, where available.

```

Listing B.1: Reporting guidelines for mass spectrometry from MIAPE-MS v2.97.8

The MIAPE-MS checklist in tab-delimited format

```

<!-- MIBBI\_Foundry\_compilation 2012-06-20T11.09.35+01.00 -->
Index no. Item name Item value Definition Examples
1. Mass\_spectrometry
1.1. General features
1.1.1. Global descriptors
1.1.1.1. Date
1.1.1.1.1. Date given as YYYY-MM-DD
1.1.1.2. Person or responsible role
1.1.1.2.1. Name Name should either be of the person or the institutional role
holding responsibility for the associated (meta)data.
1.1.1.2.2. Affiliation
1.1.1.2.3. Stable contact information Such information should be one or more from
the following list: email; telephone (plus international dialling code); postal
address.
1.1.1.2.4. Globally-unique person ID (authority and identifier) Where available,
please give a unique person identifier and the system of which it is part (e.g.,
ISNI, ResearcherID, etc.).
1.1.1.3. Instrument manufacturer and model
1.1.1.4. Customisations Summary thereof.
1.1.2. Control and analysis software
1.1.2.1. Software name and version
1.1.2.2. Switching criteria Tandem instruments only.
1.1.2.3. Isolation width Global, or by MS level
1.1.2.4. Location of parameters file
1.2. Ion sources As each spectrum is acquired using only one ionisation source,
select only the subcategory that applies.
1.2.1. Electrospray ionisation
1.2.1.1. Supply type Static, fed.

```

- 1.2.1.2. Interface manufacturer model and catalog number where available
- 1.2.1.3. Sprayer type, coating, manufacturer, model and catalogue number where available
- 1.2.1.4. Relevant voltages where appropriate Provide tip, cone, acceleration
- 1.2.1.5. Other parameters if discriminant for the experiment Nebulising gas, pressure.
- 1.2.2. MALDI
 - 1.2.2.1. Plate composition or type
 - 1.2.2.2. Matrix composition if applicable
 - 1.2.2.3. Deposition technique
 - 1.2.2.4. Relevant voltages where appropriate Grid, acceleration.
 - 1.2.2.5. PSD summary if performed Or LID/ISD.
 - 1.2.2.6. Operation with or without delayed extraction
 - 1.2.2.7. Laser type and wavelength in nm Nitrogen laser
 - 1.2.2.8. Other laser related parameters, if discriminating for the experiment Pulse energy in J, attenuation, focus diameter in m, pulse duration as ns at FWHM, frequency in Hz and average shots fired per spectrum.
- 1.3. Post-source component As an MS experiment performed on one instrument cannot be acquired using all existing analysers and detectors, select the elements that apply. N.B. Ion optics, simple quadrupoles, hexapoles no parameters to be captured. FT-ICR – As for Ion trap and Collision cell combined, no further parameters required.
 - 1.3.1. Time-of-flight drift tube
 - 1.3.1.1. Reflectron status On, off, none.
 - 1.3.2. Ion trap
 - 1.3.2.1. Final MS stage achieved
 - 1.3.3. Collision cell
 - 1.3.3.1. Gas type and pressure bar
 - 1.3.3.2. Collision energy
 - 1.3.4. Detectors
 - 1.3.4.1. Detector type
 - 1.3.4.2. Detector sensitivity
- 1.4. Spectrum and peak list generation and annotation For this section; if software other than that listed under '[Control and analysis software](#)' is used to perform a task, the producer, name and version of that software must be supplied in each case.
 - 1.4.1. Spectrum description
 - 1.4.1.1. Location of source file including file name and type The raw spectra
 - 1.4.1.2. Identifying information for the target area MALDI-like methods only.
 - 1.4.1.3. MS level for this spectrum
 - 1.4.1.4. Ion mode for this spectrum
 - 1.4.1.5. Precursor m/z and charge with the full mass spectrum containing that peak For MS level 2 and higher.
 - 1.4.2. Peak list generation
 - 1.4.2.1. Parameters triggering the generation of peak lists from raw data Include filtering for exclusion of peak lists from raw spectra, where appropriate.
 - 1.4.2.2. Acquisition number of all acquisitions combined in the peak list From the raw file – the total number combined and whether summed or averaged
 - 1.4.2.3. Smoothing Whether applied, parameters.
 - 1.4.2.4. Background threshold or algorithm used
 - 1.4.2.5. Signal-to-noise estimation and method
 - 1.4.2.6. Percentage peak height for centroiding or algorithm used if appropriate
 - 1.4.2.7. Whether charge states were calculated, spectra were deconvoluted and peaks were deisotoped With methods described as appropriate.
 - 1.4.2.8. Relative times for all acquisitions combined in the peak list Electrospray only.
 - 1.4.2.9. Base peak m-to-z where appropriate

1.4.2.10. Metastable peaks removed if applicable 1.4.2.11. m-to-z and intensity values 1.4.3. Quantitation for selected ions In addition to 'Spectrum description' and ' Peak list generation'. Only applicable if a quantitation experiment has been performed. 1.4.3.1. Experimental protocol and canonical reference where available with deviations 1.4.3.2. Number of combined samples and MS runs analysed 1.4.3.3. Quantitation approach integration 1.4.3.4. Normalisation technique 1.4.3.5. Location of quantitation data with file name and type where appropriate

Listing B.2: MIAPE-MS checklist in tab-delimited format downloaded from
<http://mibbi.sourceforge.net/MICheckout.htm>

Appendix C

Automation of the syntactic transformation process

This appendix describes the software that was written to automatically transform the XML-based data formats into candidate archetypes. First, it describes how the Java reference implementation of the *openEHR* reference model was used in the software and then it describes the steps performed by the software in the transformation process, applying the previously established mappings between the constructs of the input XML schema and output ADL files.

The Java reference implementation

As previously described, the archetype definition is an instance of a `C_COMPLEX_OBJECT` that constrains a certain *openEHR* reference model class. In the Java reference implementation this corresponds to the `CComplexObject` class (a subclass of `CObject`). `C_COMPLEX_OBJECT`s can contain `C_ATTRIBUTES` (either `C_SINGLE_ATTRIBUTE` or `C_MULTIPLE_ATTRIBUTE`), which constrain the attributes of the reference model class of the parent `C_COMPLEX_OBJECT`. The equivalent classes for these are `CAttribute`, `CSingleAttribute` and `CMultipleAttribute`. Thus at the root there is a `CObject` that can have a `CAttribute` child, that itself can have `CObject` children and so on, until the leafnodes, which constrain *openEHR* data types. To instantiate these objects, variables corresponding to their attribute values need to be set, as shown in the code snippets that show an instantiation of a `CObject` and a `CAttribute` below:

```
CComplexObject someComplexObject = new CComplexObject(objectPath , objectrmTypeName ,
    objectOccurrences , objectNodeID , attributes , objectParent );
```

These parameters need to be instantiated as follows:

- `objectPath`: this corresponds to the node path of the `CObject` object, e.g. “/data[at0002]/events[at0003]/state[at0005]/items[at0020]/items[at0024]” - this path needs to be kept track of during the creation of these nodes, so that the correct

path is assigned to each node - these paths are used for the correct serialization of the ADL file

- `objectrmTypeName`: the reference model class that this object constrains, e.g. CLUSTER or ELEMENT, as specified in mapping above
- `objectOccurrences`: occurrence parameters of this object as specified in mapping for element occurrences in XML schema/CVMappingRules
- `objectNodeID`: consecutively assigned atxxxx code
- `attributes`: list of CAttributes of this CObject
- `objectParent`: defaulted to null as it has no influence on the serialization

```
CMultipleAttribute someAttribute = new CMultipleAttribute(CAttributePath ,  
    CAttributermAttributeName , existence , cardinality , children);
```

These parameters need to be instantiated as follows:

- `CAttributePath`: this corresponds to the node path of the CAttribute object, e.g. “/data[at0002]/events[at0003]/state[at0005]/items[at0020]/items” - as for the CObjects, this path needs to be kept track of during the creation of these nodes, so that the correct path is assigned to each node - these paths are used for the correct serialisation of the ADL file
- `CAttributermAttributeName`: the attribute of the parent reference model class that this attribute constrains, e.g. data or items - since only CLUSTERS or ELEMENTs are being used, these can only be “items” (the only attribute of the CLUSTER reference model class) or “value” (attribute of the ELEMENT reference model class)
- `existence`: defaulted to required (the relationship between existence and cardinality attributes of CAttribute and of the occurrences attribute of CObject have previously been described in Chapter 2)
- `cardinality`: the cardinality intervals are dictated by the attributes specifying element occurrences in the XML schema and CVMappingRules files (specifying how many items must be in the container, and whether it acts logically as a list, set or bag (from [43]))
- `children`: list of CObjects of this CAttribute

As can be seen from the way the Java objects need to be initialized, to create an archetype definition, the archetype structure will have to be built up in a tree-like fashion, starting with the leaf-nodes over the branches down to the stem, while keeping track where each node will be positioned within the final archetype definition.

Special cases exist for leaf node classes that represent *openEHR* data types, such as the *CDvQuantity* class for the *DV_QUANTITY* data type. Each of the leaf node classes needs to be instantiated in a different way, depending on what kind of information they represent.

Further, for each *CObject* that is being created, in parallel an *term_definition* item for the archetype ontology needs to be created, using the *ArchetypeTerm* class. For example:

```
ArchetypeTerm someArchterm = new ArchetypeTerm(currentNodeID, termName, description);
```

These parameters need to be instantiated as follows:

- *currentNodeID*: this corresponds to the *atxxxx* node ID to which this term corresponds
- *termName*: the mapped name of this term from the XML schema or CV
- *description*: the mapped description of this term from the XML schema or CV

Further, for each ontology term, a term binding is established and added to the Ontology binding list it not in list yet:

```
List<List<OntologyBindingItem>> OBL = new LinkedList(new LinkedList<
    OntologyBindingItem>());

TermBindingItem obi = new TermBindingItem(currentNodeID, termList);
OBL.add(ontologyBindingList);
```

These parameters need to be instantiated as follows:

- *currentNodeID*: this corresponds to the *atxxxx* node ID to which this term corresponds
- *termList*: this contains the "terminology::termAccessionNumber" combination, "PSI-MS::1000788" for the PSI-MS term with id 1000788.

Steps performed by the automatic transformation software

The transformation software performs the following high-level steps:

1. Load and parse XSD file and auxiliary files, such as CVMappingRule and referenced CVs.
2. Discover all the xs:complexType elements defined in the XSD file.
3. For each xs:complexType element, a corresponding ADL file of type CLUSTER is created according to the mappings established between the two formats as described below.

The steps involved in the creation of a CLUSTER file are described in the next section.

Creation of the CLUSTER file for each xs:complexType object

1. Create the archetype object with default header section values
2. Creating a root CLUSTER in the archetype definition and add corresponding term to term definition in the archetype ontology, using the name of the xs:complexType element and annotation documentation given in the XML schema in the description.
3. Create child attribute of type “items”
4. Create child C_COMPLEX_OBJECTs for each xs:element or xs:attribute defined in the xs:complexType element (possible listed under an xs:sequence element) as follows:
 - (a) If child node is an xs:element node:
 - i. If type attribute is a dx:CVParamType, check for matching CVMappingRule, extract mapped ontology terms.
 - A. If ontology terms are text type terms, create corresponding CV_CODED_TEXT leaf node for text type CV terms and corresponding term definition in archetype ontology. Define and add corresponding object constraints. Bind terms to source ontology in term binding section.
 - B. If ontology terms are data type terms, create objects of type ELEMENT that constrain corresponding *openEHR* data types and corresponding term definitions in archetype ontology. Define and add corresponding object constraints. Bind terms to source ontology in term binding section.

- ii. If type is a `dx:complexType`, create slot for a CLUSTER for the archetype defined by the referred-to `xs:complexType`. Define and add corresponding object constraints.
- iii. If type is `xs:simpleType`, create an object of type ELEMENT that constrains the corresponding *openEHR* data type and corresponding term definitions in archetype ontology. Define and add corresponding object constraints.

(b) If child node is an `xs:attribute` node:

- i. Create object of type ELEMENT that constrains the *openEHR* data type corresponding to the simple type specified in the attribute and corresponding term definition in archetype ontology. Define and add corresponding object constraints.

(c) If child node is an `xs:complexContent` childnode:

- i. Create object corresponding to the `xs:element` mentioned in the base attribute of the `xs:extension` child node and create archetype objects corresponding to the attribute or (sequence of) element node(s) under the `xs(d):extension` node and corresponding term definition in archetype ontology. Define and add corresponding object constraints.

5. Serialize archetype into ADL.

Each step in the software was tested to see if the output archetypes would correctly represent the information provided in the source data format.

Later modifications of the “base” software

Later, the above “base” software program was further refined reacting on observations made during the semantic review process. For example, the default of slot creation was changed to include the structure of the referenced `dx:complexType` element directly in the parent ADL file as a CLUSTER object when the content of the candidate archetype created for the child element was evaluated to be too small to justify a stand-alone archetype. Thus, instead of having to manually copying structures around text files that may lead to inconsistencies and errors, the software was rerun using the new configurations, thus reducing the amount of copying error that may be introduced. Furthermore, some `xs:complexType` elements were found to be unsuitable as ADL equivalents and could thus be ignored, such as “ReferenceableParamGroupRefType” or “CVList” - and were thus ignored during the transformation process.

Appendix D

Archetype checkpoints

Archetype checkpoints table provided by *openEHR*¹:

Concept name	Is this appropriate?
Data - data elements	Are these complete? Is there any content missing? Are the datatypes appropriate?
Data - normal statements	Should normal statements be included in this archetype? If present, are the normal statements appropriate? What normal statements should be added?
Events	Should any event be available? Are the specific point-in-time or interval events appropriate? What specific events should be added? Are events present that only apply in limited use cases and should be left to a template?
Protocol	Are the Protocol data elements appropriate? What other data elements should be added?
State	Are the State data elements appropriate? Are the assumed values correct? What other data elements should be added?
Slots	Are the slots named appropriately? Are the ITEM archetypes selected as inclusions correct? Are the ITEM archetypes selected as exclusions correct?
Comments	Check the correctness of any comments per data element
Cardinality	Check that cardinality is correct for Compositions, Sections, Clusters and Slots
Occurrences	Check the occurrences of data elements is correct
Metadata	Check completion and correctness of: - Concept Description - Purpose - Use - Misuse - References - Citation style - Keywords - Primary Author - Contributors
Punctuation and spelling	Check for correctness and consistency of punctuation and spelling
Phrasing and expression	Check for consistency of phrasing and expression, especially in data element naming and descriptions

Table D.1: Archetype checkpoints

¹Editorial checklist, <http://www.openEHR.org/wiki/display/healthmod/Editorial+checklist+for+clinical+content+review+of+archetypes>

Appendix E

List of candidate archetypes from mzML

Candidate archetype	Content description from element documentation	Action taken
CLUSTER.mzML	This is the root element for the Proteomics Standards Initiative (PSI) mzML schema, which is intended to capture the use of a mass spectrometer, the data generated, and the initial processing of that data (to the level of the peak list).	Eliminated - container type.
CLUSTER.file.description	Information pertaining to the entire mzML file (i.e. not specific to any part of the data set) is stored here.	Absorbed into OBSERVATION.mass_spectrometry (formerly CLUSTER.run), modelled under “summary ITEM.STRUCTURE” attribute of HISTORY (While the <i>openEHR</i> reference model clearly allows for this attribute [240], the <i>openEHR</i> Archetype Editor does not validate it, therefore it has been added manually.
CLUSTER.source_file_list	List and descriptions of the source files this mzML document was generated or derived from	Eliminated - container type, used out of context for aggregation only.
CLUSTER.source_file	Description of the source file, including location and type.	Too generic and overlap with DV_MULTIMEDIA detected: absorbed into parent archetypes (CLUSTER.scan_settings, CLUSTER.mass_spectrum, CLUSTER.scan, CLUSTER.precursor and CLUSTER.run)
CLUSTER.cv	Information about an ontology or CV source and a short 'lookup' tag to refer to.	Eliminated - auxiliary content, terminology bindings modelled differently in archetypes.
CLUSTER.cv_list	Container for one or more controlled vocabulary definitions.	Eliminated - container type, used out of context for aggregation only and auxiliary content.
CLUSTER.referenceable_param_group_list	Container for a list of referenceableParamGroups	Eliminated - container type, used out of context for aggregation only.
CLUSTER.param_group	Structure allowing the use of a controlled (cv-Param) or uncontrolled vocabulary (userParam), or a reference to a predefined set of these in this mzML file (paramGroupRef).	Eliminated - content too generic, does not represent a domain-level concept.
CLUSTER.referenceable_param_group	A collection of CVParam and UserParam elements that can be referenced from elsewhere in this mzML document by using the 'paramGroupRef' element in that location to reference the 'id' attribute value of this element.	Eliminated - content too generic, does not represent a domain-level concept.
CLUSTER.cv_param	This element holds additional data or annotation. Only controlled values are allowed here.	Eliminated - auxiliary content, terminology bindings modelled differently in archetypes.
CLUSTER.user_param	Uncontrolled user parameters (essentially allowing free text). Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead	Eliminated - content too generic, does not represent a domain-level concept - use special CLUSTER with “any” value in its place where it appears in other archetypes.
CLUSTER.referenceable_param_group_ref	A reference to a previously defined ParamGroup, which is a reusable container of one or more cv-Params.	Eliminated - content too generic, does not represent a domain-level concept.
CLUSTER.sample_list	List and descriptions of samples.	Eliminated - container type, used out of context for aggregation only.
CLUSTER.sample	Expansible description of the sample used to generate the dataset, named in sampleName.	Overlap with CLUSTER.specimen detected and resolved. Content not covered in CLUSTER.specimen absorbed into CLUSTER.specimen and CLUSTER.sample eliminated.

CLUSTER.instrument_configuration_list	List and descriptions of instrument configurations. At least one instrument configuration must be specified, even if it is only to specify that the instrument is unknown. In that case, the “instrument model” term is used to indicate the unknown instrument in the instrumentConfiguration.	Eliminated - container type, used out of context for aggregation only.
CLUSTER.component		Overlap with CLUSTER.device detected and resolved, resulting in specialisation into CLUSTER.device-mass_spectrometer.
CLUSTER.source_component	This element must be used to describe a Source Component Type. This is a PRIDE3-specific modification of the core MzML schema that does not have any impact on the base schema validation.	Overlap with CLUSTER.device detected and resolved, resulting in specialisation into CLUSTER.device-mass_spectrometer-source.
CLUSTER.detector_component	This element must be used to describe a Detector Component Type. This is a PRIDE3-specific modification of the core MzML schema that does not have any impact on the base schema validation.	Overlap with CLUSTER.device detected and resolved, resulting in specialisation into CLUSTER.device-mass_spectrometer-detector.
CLUSTER.analyzer_component	This element must be used to describe an Analyzer Component Type. This is a PRIDE3-specific modification of the core MzML schema that does not have any impact on the base schema validation.	Overlap with CLUSTER.device detected and resolved, resulting in specialisation into CLUSTER.device-mass_spectrometer-analyzer.
CLUSTER.component_list	List with the different components used in the mass spectrometer. At least one source, one mass analyzer and one detector need to be specified.	Eliminated - container type, used out of context for aggregation only.
CLUSTER.instrument_configuration	Description of a particular hardware configuration of a mass spectrometer. Each configuration must have one (and only one) of the three different components used for an analysis. For hybrid instruments, such as an LTQ-FT, there must be one configuration for each permutation of the components that is used in the document. For software configuration, use a ReferenceableParamGroup element.	Overlap with CLUSTER.device-details detected and resolved, resulting in specialisation into CLUSTER.device-details-mass_spectrometer.
CLUSTER.software_ref	Reference to a previously defined software element	Eliminated - reference type archetype.
CLUSTER.software_list	List and descriptions of software used to acquire and/or process the data in this mzML file.	Eliminated - container type, used out of context for aggregation only.
CLUSTER.software	Software information.	Overlap detected with CLUSTER.device-details. Software concept extracted from CLUSTER.device-details to become stand-alone CLUSTER.software archetype. Specialized for CLUSTER.software-mass_spectrometer
CLUSTER.data_processing_list	List and descriptions of data processing applied to this data.	Eliminated - container type, used out of context for aggregation only.
CLUSTER.data_processing	Description of the way in which a particular software was used.	Too specific to mass spectrometry, content absorbed into parent archetypes (CLUSTER.mass_spectrum, CLUSTER.mass_chromatogram, OBSERVATION.mass_spectrometry).
CLUSTER.processing_method		Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.data_processing).
CLUSTER.scan_settings_list	List with the descriptions of the acquisition settings applied prior to the start of data acquisition.	Eliminated - container type, used out of context for aggregation only.
CLUSTER.scan_settings	Description of the acquisition settings of the instrument prior to the start of the run.	Too specific to mass spectrometry, content absorbed into parent archetype (CLUSTER.device-details-mass_spectrometer).
CLUSTER.target_list	Target list (or 'inclusion list') configured prior to the run.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.scan_settings)
CLUSTER.run	A run in mzML should correspond to a single, consecutive and coherent set of scans on an instrument.	Root for the description of the mass spectrometry experiment. Re-modelled into OBSERVATION.mass_spectrometry archetype (v1 without use_node and “summary” attribute to allow it to be opened in the openEHR Archetype Editor; v2 includes the use_node construct and “summary” attribute, but the openEHR Archetype Editor does not support these constructs even though the openEHR specification defines them).
CLUSTER.source_file_ref		Eliminated - reference type.
CLUSTER.source_file_ref_list	Reference to a previously defined sourceFile.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.scan).
CLUSTER.spectrum_list	List and descriptions of spectra.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (OBSERVATION.mass_spectrometry, modelled under “data” attribute).

CLUSTER.scan_window_list		Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.scan).
CLUSTER.scan_list	List and descriptions of scans.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (OBSERVATION.mass_spectrometry, modelled under "state" attribute).
Scan	Scan or acquisition from original raw file used to create this peak list, as specified in sourceFile.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.scan_list).
CLUSTER.precursor_list	List and descriptions of precursor isolations to the spectrum currently being described, ordered.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.mass_spectrum).
CLUSTER.precursor	The method of precursor ion selection and activation	Content too small and specific for stand-alone archetype. Absorbed into parent archetypes (CLUSTER.mass_chromatogram and CLUSTER.precursor_list).
CLUSTER.selected_ion_list	The list of selected precursor ions.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.precursor).
CLUSTER.product_list	List and descriptions of product isolations to the spectrum currently being described, ordered.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (CLUSTER.mass_spectrum).
CLUSTER.product	The method of product ion selection and activation in a precursor ion scan	Content too small and specific for stand-alone archetype. Absorbed into parent archetypes (CLUSTER.mass_chromatogram and CLUSTER.product_list).
CLUSTER.binary_data_array_list	List of binary data arrays.	Content too small and specific for stand-alone archetype. Absorbed into parent archetypes (CLUSTER.mass_chromatogram and CLUSTER.mass_spectrum).
CLUSTER.binary_data_array	The structure into which encoded binary data goes. Byte ordering is always little endian (Intel style). Computers using a different endian style must convert to/from little endian when writing/reading mzML	Content modelled with DV_PARSABLE data type.
CLUSTER.spectrum	The structure that captures the generation of a peak list (including the underlying acquisitions). Also describes some of the parameters for the mass spectrometer for a given acquisition (or list of acquisitions).	OK.
CLUSTER.chromatogram_list	List of chromatograms.	Content too small and specific for stand-alone archetype. Absorbed into parent archetype (OBSERVATION.mass_spectrometry).
CLUSTER.chromatogram	A single chromatogram.	OK.

Table E.1: List of mzML candidate archetypes

Appendix F

List of candidate archetypes from FuGE

Candidate archetype	Content description from element documentation	Action taken
CLUSTER.sequence_annotation_set	This is used to annotate a set of sequences that share the same species, type and PolymerType.	Too specific for stand-alone archetype, concept overlaps with CLUSTER.sequence and contents were combined into CLUSTER.sequence.
CLUSTER.sequence	A Sequence is a representation of a DNA, RNA, or protein sequence.	Modified to become CLUSTER.biological_molecule_sequence archetype.
CLUSTER.conceptual_molecule	An abstract class for describing database entries of biological molecules such as DNA and protein sequences, metabolites or lipids etc.	Eliminated as content too generic, abstract super class to CLUSTER.sequence - concept already covered in CLUSTER.sequence archetype.
CLUSTER.investigation	An Investigation is a self-contained unit of study that contains one or more types of 'omics technology.	Modified to become EVALUATION.omic_investigation archetype.
CLUSTER.investigation_component	An overview of the design using a single omics technique.	Modified to become OBSERVATION.omic_experiment archetype.
CLUSTER.factor	Factors are the independent variables of an experiment (e.g. time, glucose concentration, ...). Note that Factors can be shared by multiple InvestigationComponents.	Concept too small - absorbed into CLUSTER.investigation_component to replace non-composite association.
CLUSTER.factor_value	The value for a Factor.	Concept too small - absorbed into CLUSTER.factor
CLUSTER.higher_level_analysis	A container for domain-specific analyses performed on data. Subclasses must be created for different domains.	Conent absorbed into CLUSTER.investigation.
CLUSTER.data	Data can be an input to or an output from a ProtocolApplication. Data may be produced from a Material (data acquisition) or from another Data object (data transformation). Examples of Data are gene expression measurements, or phenotypes associated with genetic manipulations.	Eliminated - too generic, use <i>openEHR</i> data types to describe data (replaced by "any" where needed).
CLUSTER.dimension	A single dimension of a coordinate system. The DimensionElement class represents the discrete domain values represented in this dimension. A dimension with infinite domain values is not modeled here, for practical reasons.	Concept too small - absorbed into CLUSTER.factor.
CLUSTER.dimension_element	A single point within a parent dimension. This class should be extended.	Concept too small - absorbed into CLUSTER.factor.
CLUSTER.external_data	Data external to the XML instance document. The location of the data file is given in the location attribute.	Eliminated - too generic, use DV_MULTIMEDIA instead.
CLUSTER.internal_data	Data that is internal to an instance XML document. InternalData can be extended with an element that defines a particular encoding or data type for the storage array or the subclass GenericInternalData should be instantiated. The array will typically use pointer arithmetic to access values based on the rank (e.g. number of) Dimensions and their respective sizes (e.g. the number of contained DimensionElements).	Eliminated - definition specific to XML technology, use <i>openEHR</i> data types to describe data (replaced by "any" where needed).
CLUSTER.data_partition	An abstract class to represents a segmentation of the data by indexing existing DimensionElements of a Data. The class can be extended to reference specific subclasses of DimensionElement.	Eliminated - definition specific to XML technology and too generic.
CLUSTER.partition_pair	Relates a DataPartition (subset of Data) from one data set to another data set, for example for "supporting evidence".	Eliminated - definition specific to XML technology and too generic.

CLUSTER.generic_internal_data	Data that is internal to an instance XML document stored in an array with a specified encoding. The array will typically use pointer arithmetic to access values based on the rank (e.g. number of) Dimensions and their respective sizes (e.g. the number of contained DimensionElements).	Eliminated - definition specific to XML technology, use <i>openEHR</i> data types to describe data (replaced by “any” where needed).
CLUSTER.generic_data_partition	Represents a segmentation of the data by indexing existing DimensionElements of a Data	Eliminated - definition specific to XML technology, use <i>openEHR</i> data types to describe data (replaced by “any” where needed).
CLUSTER.material	Material represents any kind of substance used in an experimental workflow, such as whole organisms, cells, DNA, solutions, compounds and experimental substances (gels, arrays etc.). The Material class can be extended by adding subclasses to model domain specific properties, or the relationships to OntologyIndividual can be used to describe the characteristics and type of Material. Materials can be related to other materials through a directed acyclic graph (represented by ProtocolApplication(s)). Sub-component materials can be represented by the self-association on Material (...). These associations are abstract and should be extended to represent these semantics for extensions of ProtocolApplication and Material.	Combined with CLUSTER.generic_material.
CLUSTER.material_measurement	MaterialMeasurement represents the quantity of a source Material used in a ProtocolApplication. MaterialMeasurement is abstract and should be extended if a subclass of ProtocolApplication has to be associated with a measured source of a subclass of Material.	Eliminated - concept too small, content absorbed into CLUSTER.material.
CLUSTER.generic_material	A subclass of the abstract Material class, which should be used in conjunction with controlled vocabulary terms to describe Materials of any types used in an investigation.	Combined with CLUSTER.material.
CLUSTER.generic_material_measurement	A subclass of MaterialMeasurement to be used without being extended in conjunction with GenericProtocolApplication and GenericMaterial to model measured sources of materials.	Content too generic - absorbed into CLUSTER.generic_material
CLUSTER.person	A person for which the attributes are self describing.	Eliminated, overlaps with CLUSTER.individual_personal.
CLUSTER.security	Permission information for an object as to ownership, write and read permissions.	Eliminated - security covered in <i>openEHR</i> specification, not modelled on domain level.
CLUSTER.audit	Tracks information on the Contact that creates or modifies an object.	Eliminated - audit covered in <i>openEHR</i> specification, not modelled on domain level.
CLUSTER.organization	Organizations are entities like companies, universities, government agencies for which the attributes are self describing.	Eliminated, overlaps with CLUSTER.organization.
CLUSTER.security_group	Groups Contacts together based on their security privileges.	Eliminated - security layer covered in <i>openEHR</i> specification, not modelled on domain level.
CLUSTER.contact	A contact is either a person or an organization.	Eliminated, overlaps with CLUSTER.organization and CLUSTER.individual_professional.
CLUSTER.contact_role	The role that a Contact plays in an organization or with respect to the associating class. A Contact may have several Roles within scope, and as such, associations to ContactRole allow the use of a Contact in a certain manner. Examples might include a provider, or a data analyst.	Eliminated, overlaps with CLUSTER.professional_role
CLUSTER.security_access	A specification of the access right (e.g. read or write) that a specified group (SecurityGroup) has on a particular object.	Eliminated - security layer covered in <i>openEHR</i> specification, not modelled on domain level.
CLUSTER.bibliographic_reference	Represents bibliographic references, including the most common attributes. Note that because a BibliographicReference is Identifiable, a DatabaseEntry can also be specified.	OK.
CLUSTER.database	An address to a repository.	OK.
CLUSTER.database_reference	A reference to a record in a database.	OK.
CLUSTER.description	A free text description of an object.	Eliminated - use <i>openEHR</i> data type DV.TEXT instead.
CLUSTER.URI	A URI is short for Uniform Resource Identifier. A URI is a compact sequence of characters that identifies an abstract or physical resource.	Eliminated - use <i>openEHR</i> data type DV.URI instead.
CLUSTER.name_value_type	A tuple designed to store data, keyed by a name and type.	Eliminated - too generic, modelled with internal structures and data types in <i>openEHR</i> archetypes.
CLUSTER.protocol	A Protocol is a parameterizable description of a method. ProtocolApplication is used to specify the ParameterValues of its Protocol's Parameters. Protocol should be extended in data formats. For cases where no extension is developed, the subclass of Protocol, GenericProtocol, should be used to capture experimental protocols.	Combined with CLUSTER.protocol_application and CLUSTER.action into CLUSTER.protocol.

CLUSTER.parameter	A Parameter is a replaceable value in a Parameterizable class. Examples of Parameters include: scanning wavelength, laser power, centrifuge speed, multiplicative errors, the number of input nodes to a SOM, and PCR temperatures. Parameter is abstract and should be extended by subclassing. The GenericParameter class offers the functionality of a parameter defined by a controlled vocabulary term.	Eliminated - too generic, use appropriate <i>openEHR</i> data types instead.
CLUSTER.parameter_value	The value of a Parameter.	Eliminated - too generic, use appropriate <i>openEHR</i> data types instead.
CLUSTER.protocol_application	The use of a protocol with the requisite Parameters and ParameterValues. ProtocolApplications can take Material or Data (or both) as input and produce Material or Data (or both) as output. ProtocolApplication is abstract and should be subclassed in the development of modular formats. The subclass GenericProtocolApplication can be used without extension.	Concept overlap with CLUSTER.protocol, absorbed into CLUSTER.protocol.
CLUSTER.software	Software used in the Protocol. Examples of Software include: feature extraction software, clustering software, etc... Software is abstract and should either be extended by subclassing or the GenericSoftware class, a functional version of Software, should be used without extension.	Overlap with CLUSTER.software archetype developed during development of archetypes for mass spectrometry - included new items into CLUSTER.software.
CLUSTER.equipment	The equipment (hardware) used in the Protocol. Examples include: computers, scanners, wash stations etc... Equipment is abstract and should either be extended by subclassing or the GenericEquipment class, a functional version of Equipment, should be used.	Eliminated - overlaps with CLUSTER.device archetype.
CLUSTER.equipment_application	The use of a piece of Equipment with the requisite Parameters and ParameterValues.	Eliminated - overlaps with CLUSTER.device archetype.
CLUSTER.software_application	The use of a piece of software with the requisite Parameters and ParameterValues.	Eliminated - overlaps with CLUSTER.software which already covers all items.
CLUSTER.parameterizable	The Parameterizable interface encapsulates the association of Parameters with ParameterValues.	Eliminated - too generic, use appropriate <i>openEHR</i> data types instead.
CLUSTER.parameterizable_application	The interface that is the use of a Parameterizable class.	Eliminated - too generic.
CLUSTER.action	An Action is one step of a Protocol. Sets of ordered Actions define the Protocol. Action is abstract and can be extended to specify particular types of steps within a subclass of Protocol.	Concept overlap with CLUSTER.protocol, absorbed into CLUSTER.protocol.
CLUSTER.action_application	ActionApplication mirrors the structure of Actions within Protocols and allows ActionDeviations to be reported. RULE: If an Action references a childProtocol, an ActionApplication must be created and reference a childProtocolApplication of the corresponding type (or GenericProtocolApplication) to mirror the Protocol structure. ActionApplications can be (but need not be) created for simple Actions.	Concept overlap with CLUSTER.action, absorbed into CLUSTER.action.
CLUSTER.generic_protocol	GenericProtocol should be used with GenericAction, GenericParameter or protocolText to describe protocols in a data format for cases where no explicit extension of Protocol has been developed.	Concept overlap with CLUSTER.protocol, absorbed into CLUSTER.protocol.
CLUSTER.generic_action	A GenericAction represents a step within a GenericProtocol. It allows a reference to a sub-GenericProtocol, user entered text to describe the GenericAction or a term from a controlled vocabulary to be given.	Concept overlap with CLUSTER.action, absorbed into CLUSTER.action.
CLUSTER.generic_software	A subclass of the abstract Software class for capturing the description of Software used.	Eliminated - CLUSTER.device_details contains slot for CLUSTER.software.
CLUSTER.generic_equipment	A subclass of the abstract Equipment class for capturing the description of Equipment used.	Eliminated - overlaps with CLUSTER.device archetype.
CLUSTER.generic_parameter	A subclass of the abstract Parameter class to represent a parameter that is defined by a controlled vocabulary term.	Eliminated - too generic, use appropriate <i>openEHR</i> data types instead.
CLUSTER.generic_protocol_application	A subclass of ProtocolApplication to be used as it is in data formats without being extended.	Concept overlap with CLUSTER.generic_protocol, absorbed into CLUSTER.generic_protocol.
CLUSTER.parameter_pair	A pairing of an output parameter from a Protocol (SourceParameter) with an input parameter to a separate Protocol (TargetParameter) to indicate that the ParameterValue will be the same.	Concept too small - absorbed into parent CLUSTER.generic_action archetype.
CLUSTER.ontology_individual	An instance of an Ontology class.	Eliminated - too generic, modelled with internal structures and data types in <i>openEHR</i> archetypes.
CLUSTER.ontology_property	Base class for instances of Ontology properties	Eliminated - too generic, modelled with internal structures and data types in <i>openEHR</i> archetypes.
CLUSTER.data_property	An instance of an Ontology data property. So an individual of Age might be restricted to a value of 8.	Eliminated - too generic, modelled with internal structures and data types in <i>openEHR</i> archetypes.

CLUSTER.object_property	An instance of an Ontology object property that restricts the parent individual to its child individual. So an Age individual may be restricted to an Individual unit of years.	Eliminated - too generic, modelled with internal structures and data types in <i>openEHR</i> archetypes.
CLUSTER.ontology_term	A single entry from an ontology or a controlled vocabulary. ()	Eliminated - terminology binding done differently in <i>openEHR</i> archetypes. Ontology terms may be modelled within archetype if they form part of the concept, or may be included at run-time using a [ac-NNNN] node as local constraint to DV_CODED_TEXT data type.
CLUSTER.ontology_source	The source ontology or controlled vocabulary list that ontology terms have been obtained from.	Eliminated - terminologies used for binding terms to archetype nodes are listed under the “terminologies_available” section in an archetype.
CLUSTER.describable	Abstract class that allows subclasses to inherit associations that enable objects to be given a text description, additional annotations as Ontology entries, a URI, associations to Audit for tracking changes, and the association to Security for indicating permissions. The URI, Annotations, PropertySets and Descriptions associations inherited from Describable should only be used to capture additional information where the model contains no other structures that could be used to capture the information.	FuGE base type, eliminated - too generic.
CLUSTER.identifiable	Other classes in the model can be specified as sub-classes, inheriting from Identifiable. Identifiable gives classes a unique identifier within the scope and a name that need not be unique. Identifiable also provides a mechanism for annotating objects with BibliographicReference(s) and DatabaseEntry(s).	FuGE base type, eliminated - too generic.
CLUSTER.atomic_value	An atomic value i.e. one that has a single value.	Eliminated - too generic, use built-in data types instead (e.g. C_DV_QUANTITY), depending on type of data.
CLUSTER.boolean_value	A Boolean value.	Eliminated - use built-in data type instead (DV_BOOLEAN).
CLUSTER.complex_value	A complex default value for the Parameter, such as a term from a controlled list or a function.	Eliminated - too generic, use appropriate <i>openEHR</i> data type instead (context-dependent, e.g. use DV_CODED_TEXT for choice of terms).
CLUSTER.range	A range value.	Eliminated - use built-in data type instead (DV_INTERVAL<DV_QUANTITY>)
CLUSTER.measurement	Abstract superclass representing different methods of supplying a value or measurement.	Eliminated - too generic, use built-in data type instead (e.g. C_DV_QUANTITY).
CLUSTER.protocol_collection	The complete set of Protocol, Equipment, Software and ProtocolApplication objects in the FuGE document.	Eliminated - simple collection type.
CLUSTER.investigation_collection	The collection of Investigation and Factor objects in the FuGE document.	Eliminated - simple collection type.
CLUSTER.referenceable_collection	The collection of objects that allow external references.	Eliminated - simple collection type.
CLUSTER.data_collection	The complete collection of all Data, Dimension, DataPartition and HigherLevelAnalysis objects.	Eliminated - simple collection type.
CLUSTER.conceptual_molecule_collection	The complete set of ConceptualMolecule objects and SequenceAnnotationSet objects.	Eliminated - simple collection type.
CLUSTER.ontology_collection	The complete set of OntologyIndividual and OntologySource objects referenced in the FuGE document.	Eliminated - simple collection type-/terminologies dealt with differently in <i>openEHR</i> .
CLUSTER.material_collection	The complete set of all Material objects in the FuGE document.	Eliminated - simple collection type.
CLUSTER.audit_collection	The collection of Audit objects (Security, SecurityGroup and Contacts).	Eliminated - simple collection type-/audit trail dealt with differently in <i>openEHR</i> .
CLUSTER.provider	The provider of the FuGE document in terms of the Contact and the software the produced the document instance.	Eliminated - provider of document is modelled as part of archetype class.
CLUSTER.FuGE	The root container for accessing all objects.	Eliminated - too generic, only contains container types.

Table F.1: List of FuGE candidate archetypes

Appendix G

List of candidate archetypes from GelML

Candidate archetype	Content description from element documentation	Action taken
CLUSTER.gel_ml_material_collection	Collection class for accessing all GenericMaterial instances used in GelML.	Eliminated - simple collection type.
CLUSTER.gel_ml_data_collection	Collection class for accessing all ExternalData instances in GelML.	Eliminated - simple collection type.
CLUSTER.gel_ml	The root of the GelML document.	Eliminated - root type acts as container for the collections and thus scope is too small.
CLUSTER.gel_2d_experiment	Gel2DExperiment captures the ProtocolApplications that should be reported for 2D gel electrophoresis in a GelML file.	Scope overlap - combined with CLUSTER.other_gel_experiment.
CLUSTER.gel_ml_protocol_collection	Collection class for accessing all Software, Protocol, Equipment and GenericProtocolApplications.	Eliminated - simple collection type.
CLUSTER.gel_experiment	Abstract superclass representing the groups of ProtocolApplications that should be represented in a valid GelML document.	Scope overlap - combined with CLUSTER.other_gel_experiment.
CLUSTER.gel_1d_experiment	Gel1DExperiment captures the ProtocolApplications that should be reported for 1D gel electrophoresis in a GelML file.	Scope overlap - combined with CLUSTER.other_gel_experiment.
CLUSTER.other_gel_experiment	OtherGelExperiment captures the ProtocolApplications that should be reported for a type of gel electrophoresis other than 1D or 2D in a GelML file.	Modified to become OBSERVATION.gel_electrophoresis_experiment.
CLUSTER.relative_volume	The volume of a substance relative to the medium in which it is in.	Eliminated as scope too small - can use internal archetype structure and data types instead, such as C.DV_QUANTITY to define the relative volume.
CLUSTER.volume_function	VolumeFunction represents a specification of a volume that changes, for example, over time. The value provided for a VolumeFunction should use the ComplexValue specification to represent the function.	Eliminated as scope too small - can use internal archetype structure and data types instead, such as DV_TEXT to describe volume function.
CLUSTER.absolute_volume	An absolute volume parameter.	Eliminated as scope too small - can use internal archetype structure and data types instead, such as C.DV_QUANTITY to define the relative volume.
CLUSTER.volume_parameter	Abstract superclass representing the different formats for representing the volume of a substance.	Eliminated as scope too small - can use internal archetype structure and data types instead, such as C.DV_QUANTITY to define the relative volume.
CLUSTER.substance_action	An action representing the use of a substance within a protocol. The substance can be named by free text or using the OntologyTerm association. The attribute actionText can be used to specify how the substance has been used within the protocol.	Scope too small - content absorbed into parent archetypes CLUSTER.substance_mixture_protocol and CLUSTER.detection_protocol.
CLUSTER.temporal_substance_action	The use of a substance within a protocol at a particular time point or for a particular duration.	Scope too small - content absorbed into parent archetypes CLUSTER.substance_mixture_protocol and CLUSTER.detection_protocol.

CLUSTER.substance_mixture_protocol	A Protocol representing the creation of a mixture of substances, for example to be used for specifying the components of buffers, solutions etc. If the actionText attribute of SubstanceAction is not used, the SubstanceMixtureProtocol represents only the components of the mixture and not the processes applied to them.	OK - renamed CLUSTER.substance_mixture.
CLUSTER.duration	A duration parameter. Note: values for absolute or relative time can be specified using the Parameter DefaultValue model.	Eliminated - use internal archetype structure and data types instead: DV_INTERVAL<DV_DATE_TIME>.
CLUSTER.time_point	A single time point. Note: values for absolute or relative time can be specified using the Parameter DefaultValue model.	Eliminated - use internal archetype structure and data types instead: DV_DATE_TIME.
CLUSTER.time_parameter	Abstract superclass representing Duration and TimePoints.	Content combined with sub-classes CLUSTER.duration and CLUSTER.time_point.
CLUSTER.concentration	The Concentration of a substance.	Eliminated as scope too small - can use internal archetype structure and data types instead, such as C.DV_QUANTITY to define the relative volume.
CLUSTER.mass	The mass of a substance.	Eliminated as scope too small - can use internal archetype structure and data types instead, such as C.DV_QUANTITY to define the relative volume.
CLUSTER.gel_2d	Gel2D is the material produced from a Gel2DApplication representing the combination of the two gel substrates and the proteins that have been separated.	Eliminated as overlapping CLUSTER.electrophoresed_gel can be used for this.
CLUSTER.gel_2d_protocol	A protocol describing the procedure of 2D gel electrophoresis.	Scope overlap - combined into corresponding CLUSTER.gel_2d_application archetype.
CLUSTER.gel_2d_application	Gel2DApplication is the application of a Gel2DProtocol. An instance of Gel2DApplication should reference SampleLoadingApplication, ElectrophoresisApplication, DetectionApplications and GenericProtocolApplication as its sub-steps, as defined by the referenced Gel2DProtocol.	Scope overlap with CLUSTER.gel_1d_application and CLUSTER.other_gel_application - all combined into on archetype and then absorbed into CLUSTER.other_gel_experiment.
CLUSTER.first_dimension_action	An action to represent the order of the referenced ElectrophoresisProtocol, with respect to other Actions performed within this parent protocol.	Scope too small and specific - absorbed into parent archetype CLUSTER.gel_2d_protocol.
CLUSTER.second_dimension_action	An action to represent the order of the referenced ElectrophoresisProtocol, with respect to other Actions performed within this parent protocol.	Scope too small and specific - absorbed into parent archetype CLUSTER.gel_2d_protocol.
CLUSTER.detection_action	An action to represent the order of the referenced DetectionProtocol, with respect to other Actions performed within this parent protocol.	Scope too small and specific - absorbed into parent archetypes CLUSTER.gel_2d_protocol, CLUSTER.gel_1d_protocol and CLUSTER.other_gel_protocol
CLUSTER.gel_1d	Gel1D is the material produced from a Gel1DApplication representing the combination of the gel substrate with the proteins that have been separated. Gel1D should be referenced by a DetectionApplication to describe any detection agents that have been applied.	Eliminated as overlapping CLUSTER.electrophoresed_gel can be used for this.
CLUSTER.gel_1d_application	The application of a Gel1DProtocol. Gel1DApplication should reference (via ActionApplication) SampleLoadingApplication, ElectrophoresisApplication and DetectionApplication matching the structure of the referenced Gel1DProtocol.	Scope overlap with CLUSTER.gel_2d_application and CLUSTER.other_gel_application - all combined into on archetype and then absorbed into CLUSTER.other_gel_experiment.
CLUSTER.gel_1_protocol	A protocol describing the procedure of 1D gel electrophoresis.	Scope overlap = combined into corresponding CLUSTER.gel_1d_application archetype.
CLUSTER.detection_application	Abstract superclass to represent applications of a DetectionProtocol. Its subclasses DirectDetection and IndirectDetection capture the application of the procedure. If two stainings are done, the destaining should be the first stage of the second staining process, as represented in the DetectionProtocol.	OK.
CLUSTER.detection_medium	The medium on which proteins are detected in processes such as autoradiography (example photographic film). The type of material should be captured by the inherited MaterialType association.	Scope too small - content absorbed into parent archetype CLUSTER.indirect_detection.

CLUSTER.detection_protocol	The process by which proteins are to be detected on a gel, either by a direct process e.g. staining or by an indirect process e.g. Western blot. The type of protocol should be specified using the types association inherited from Protocol	Combined with corresponding CLUSTER.detection_application.
CLUSTER.direct_detection	DirectDetection represents the application of a DetectionProtocol in which a detection agent (stain or label) is applied directly to a gel.	Scope overlap - combined with CLUSTER.detection_application.
CLUSTER.indirect_detection	IndirectDetection represents the application of a DetectionProtocol in which a detection agent (stain or label) is applied to a TransferMedium (e.g. a membrane) following proteins being transferred from a gel. Subsequently, the proteins may be detected on a DetectionMedium, such as photographic film.	Scope overlap - combined with CLUSTER.detection_application.
CLUSTER.transfer_medium	The medium proteins are transferred to during Indirect detection of proteins, such as the membrane in a Western blot. The type of material should be captured by the inherited MaterialType association.	Scope too small - content absorbed into parent archetype CLUSTER.indirect_detection.
CLUSTER.electrophoresis_step	The protocol for one step within an ElectrophoresisProtocol. The referenced parameters capture the details of the step.	Scope too small - content absorbed into parent archetype CLUSTER.electrophoresis_protocol.
CLUSTER.electrical_parameter	A class to represent the current, voltage, power or volthours etc. in each step.	Scope too small - content can be represented in parent archetype with internal ADL structure and <i>openEHR</i> data types, e.g. C.DV.QUANTITY.
CLUSTER.electrophoresis_action	A reference to the ElectrophoresisProtocol performed within the parent protocol.	Scope too small - content absorbed into parent archetypes CLUSTER.gel_ld_protocol.
CLUSTER.electrophoresis_application	An application of the referenced ElectrophoresisProtocol.	OK - renamed CLUSTER.electrophoresis.
CLUSTER.electrophoresis_protocol	A protocol for capturing the details of electrophoresis. The running conditions applied to the gel should be given in terms of voltages versus time/kilovolt hours, (or appropriate measurements) and temperature. The voltage mode profile (eg. step and hold, or gradient) should be given as ontology terms on ElectricalParameter.	Scope overlap - combined with CLUSTER.electrophoresis_application.
CLUSTER.temperature	Temperature parameter.	Scope too small - content can be represented in parent archetype with internal ADL structure and <i>openEHR</i> data types, e.g. C.DV.QUANTITY.
CLUSTER.add_buffer_action	An action to represent the use of buffers in the Protocol by reference to a SubstanceMixtureProtocol that represents the buffer and optionally the method of producing the buffer.	Scope too small - content absorbed into parent archetypes (CLUSTER.electrophoresis_protocol, CLUSTER.detection_protocol and CLUSTER.sample_loading_protocol).
CLUSTER.boundary_chain	A boundary chain is a notation for specifying an arbitrarily shaped location on an image. A starting point is given (startXCoordinate and startYCoordinate) and a set of directional steps. See chainCode documentation for this encoding.	Scope too small - content absorbed into parent archetype CLUSTER.gel_location.
CLUSTER.circle	Circle is for defining a circular location on an image.	Scope too small - content absorbed into parent archetype CLUSTER.gel_location.
CLUSTER.excised_sample	A sample excised from a gel. These materials can be referenced by formats modelling other techniques, such as mass spectrometry.	Scope too small - content absorbed into parent archetype CLUSTER.excision_application.
CLUSTER.excision_application	The Application of a protocol that describes the excision of locations on gels.	OK - made more generally reusable by replacing gel specific references with more generic references.
CLUSTER.feature_location	Abstract superclass representing the different types of method that could be used to identify a location.	Scope too small - content absorbed into parent archetype CLUSTER.gel_location.
CLUSTER.gel_location	A single location on a gel.	OK - made more generally reusable by deleting references to gel and renaming it to CLUSTER.feature_location.
CLUSTER.location_measure	A measured value, property or annotation about a GelLocation, such as predicted MW or pI.	Scope too small - content absorbed into parent archetype CLUSTER.gel_location.
CLUSTER.location_set	LocationSet represents a set of locations on a gel image.	Scope too small - content absorbed into parent archetype CLUSTER.excision_application.
CLUSTER.rectangle	Rectangle is for defining a rectangular location on an image. The X/Y coordinates correspond with the upper left corner of the rectangle.	Scope too small - content absorbed into parent archetype CLUSTER.gel_location.

CLUSTER.boundary_point_set	A set of boundary points defining a location on a gel. The BoundaryPoints are inclusive i.e. each point defines the boundary and part of the location.	Scope too small - content absorbed into parent archetype CLUSTER.gel.location.
CLUSTER.point	A single boundary point used in a BoundaryPointSet.	Scope too small - content absorbed into parent archetype CLUSTER.boundary_point_set.
CLUSTER.electrophoresed_gel	Abstract superclass for describing gels following electrophoresis.	OK.
CLUSTER.gel	The gel matrix used in any dimension of an electrophoresis assay.	OK.
CLUSTER.gel_lane	A lane on a gel.	Scope too small - content absorbed into parent archetype CLUSTER.gel.
CLUSTER.gel_manufacture_application	The application of a protocol that defines how a gel was produced.	OK - made more generally reusable by replacing gel specific references with more generic references and renaming archetype to CLUSTER.manufacture.
CLUSTER.dimensions	The measurements must be in the form of the Cartesian Coordinate system (x,y,z). According to the standard image orientation described in section 7.1.6 of the MIAPE document x represents the distance from the anode (+) to the cathode (-). For example in an IPG strip x = the strip length, for a standard slab gel, x = the width and z = the matrix depth. All dimensions should be given in milimetres.	Eliminated - overlaps with existing archetype CLUSTER.dimensions from openEHR repository.
CLUSTER.separation_range	The separation range of the gel. This can use the RangeValue or ComplexValue specification.	Scope too small - content absorbed into parent archetype CLUSTER.electrophoresed_gel.
CLUSTER.ph_range	The estimated or known pH range of a gel that separates substances based on their isoelectric point.	Scope too small - combined with archetype for abstract superclass CLUSTER.separation_range.
CLUSTER.mol_weight_range	The estimated or known range of a gel that separates substances by their molecular weight.	Scope too small - combined with archetype for abstract superclass CLUSTER.separation_range.
CLUSTER.other_gel_range	A gel range other than molecular weight or pH by which proteins are separated.	Scope too small - combined with archetype for abstract superclass CLUSTER.separation_range.
CLUSTER.acrylamide_to_cross_linker	Describes the acrylamide:crosslinker ratio, where the crosslinker type is given by an ontology term (e.g. bisacrylamide). The values stored in the attributes can be represented as a simple ratio (e.g. 37.5:1) or as percentages out of 100 (97:3). If a monomer other than acrylamide has been used, this object should not be used and the components should be specified as additional substances.	Scope too small - content absorbed into parent archetype CLUSTER.gel.
CLUSTER.measured_material	A subclass of Material for associating a quantity with a material description.	Scope overlap - content combined into CLUSTER.material archetype.
CLUSTER.sample_loading_action	An action to represent the order of the referenced SampleLoadingProtocol, with respect to other Actions performed within this parent protocol.	Scope too small - content absorbed into parent archetypes CLUSTER.other_gel_protocol, CLUSTER.gel_1d_protocol and CLUSTER.gel_2d_protocol.
CLUSTER.sample_loading_application	An application of the SampleLoadingProtocol. SampleLoadingApplication defines how a sample was loaded to a gel. To capture loading a first dimension gel onto a second dimension gel, GenericMaterialMeasurement should reference the first dimension gel and SampleLoadingApplication references the second dimension Gel (as LoadedGel).	OK - made more generically reusable by replacing gel specific references with more generic references. Renamed CLUSTER.sample_loading.
CLUSTER.sample_loading_protocol	A Protocol defining how samples are loaded onto a Gel or GelLanes.	Scope overlap - combined into CLUSTER.sample_loading_application.
CLUSTER.inter_dimension_action	The Action represent a reference to a Protocol for the stages that occur between electrophoresis over particular dimensions and an ordering of the reference with respect to other Actions performed within the parent Protocol.	Scope too small - content absorbed into parent archetypes CLUSTER.gel_2d_protocol and CLUSTER.other_gel_protocol.
CLUSTER.other_gel	The Gel output from an OtherGelApplication representing the combination of gel substrates and proteins separated.	Scope overlap - combined into CLUSTER.electrophoresed_gel.
CLUSTER.other_gel_application	OtherGelApplication is the application of an OtherGelProtocol. An instance of OtherApplication should reference (via ActionApplication) SampleLoadingApplication, ElectrophoresisApplication, DetectionApplication and GenericProtocolApplication as its sub-steps, as defined by the referenced OtherGelProtocol.	Scope overlap - combined into CLUSTER.other_gel_experiment.

CLUSTER.other_gel_protocol	A protocol for capturing experiments that cannot be represented by Gel1DProtocol or Gel2DProtocol.	Scope overlap - combined into corresponding CLUSTER.other_gel_application archetype.
CLUSTER.dimension_action	An action to represent the order of the referenced ElectrophoresisProtocol, with respect to other Actions performed within this parent protocol.	Scope too small - content absorbed into parent archetype CLUSTER.other_gel_protocol.
CLUSTER.image	A reference to an image file stored in an external location.	OK.
CLUSTER.image_acquisition_application	An application of a protocol (GenericProtocol) describing the scanning of a material to produce an Image. In this context it is expected that the Material would be an ElectrophoresedGel, TransferMedium or DetectionMedium.	OK - made more generally reusable by deleting references to gel specific content, renamed CLUSTER.image_acquisition.
CLUSTER.image_acquisition_protocol	A protocol describing image acquisition.	Content combined into CLUSTER.image_acquisition_application.
CLUSTER.auto_calibration	A parameter to represent whether auto-calibration was performed. This should use BooleanValue (subclass of Measurement).	Scope too small - content absorbed into parent archetype CLUSTER.image_acquisition_protocol.

Table G.1: List of GelML candidate archetypes

Appendix H

openEHR system implementations

Introduction

The thesis is concerned with the modelling of archetypes and templates for the eBMRR and not the implementation of a full *openEHR* system. Archetypes are employed at run-time through templates to allow the creation of an archetype-based user interface that can be used for data input to create instances of these archetypes, which can then be stored in a backend database. Due to the size and complexity of the specification, the development effort required to create even a minimal *openEHR* capable system with basic functionality is very time-consuming and beyond the scope of this thesis.

The *openEHR* specification specifies a complete EHR system. An overview of all the components necessary to build a complete *openEHR*-based EHR system has been provided by *openEHR* in a roadmap for the *openEHR* Java Reference Implementation Project¹. Briefly, the architecture of the system is divided into a domain layer (containing the kernel components, service components and wrappers), persistence layer and presentation layer (providing a screen form generator and validator). Due to the two-level modelling approach, the domain layer is more complex than that of any conventional, single-level approaches to software development.

Existing *openEHR* system implementations

Ocean Informatics are the main provider of commercial implementations of *openEHR* systems², but a request to use their system was denied.

However, some open-source prototype implementations of (parts of) an *openEHR* system exist.

For example, there are two projects that implement basic kernel functionality of

¹Roadmap, <http://www.openEHR.org/wiki/display/projects/Roadmap+of+the+openEHR+Java+project>

²List of commercial implementations: <http://www.openEHR.org/shared-resources/usage/commercial.html>

the domain layer, one in Python³ and one in Java 1.5⁴ [257, 76]. They provide the basis for some of the prototype system implementations.

opereffa

The opereffa (*open*EHR REference Framework and Application) project⁵ is an open-source implementation that is build on top of the open source Java reference implementation of the *open*EHR reference model describe in Section H. The system is at a very early stage and existing functionalities are basic and need to be improved upon.

EHRFlex

Another open-source implementation of an archetype based EHR system is EHRflex⁶. The prototype does not provide template functionality.

openEHR-Gen Framework

This open-source framework⁷ Development of this EHR system took 16 months⁸. It reuses parts of the *open*EHR Java reference implementation. The framework only constitutes the core of an EHR system, not a whole system.

openEHRApp

*open*EHRApp is a Japanese open source EHR Application, developed by a group of international students in Kanolab, Waseda University in Tokyo, Japan⁹. No documentation available or could be retrieved by contacting Kanolab.

Comparison of the implementations

Table H.1 below compares the above described *open*EHR implementations with regard to reference model used, template support, available documentation and system completeness.

While each system is incomplete, only the openEHR-Gen Framework and openEHRApp support *open*EHR archetypes. Furthermore, the openEHR-Gen Framework provides

³*open*EHR kernel in Python, <http://www.openEHR.org/projects/python.html>

⁴*open*EHR kernel in Java <http://www.openEHR.org/projects/java.html>

⁵opereffa, <http://www.openEHR.org/projects/opereffa.html>, <http://opereffa.chime.ucl.ac.uk/introduction.jsf>

⁶EHRflex, <http://ehrflex.sourceforge.net/>

⁷openEHR-gen Framework, <http://code.google.com/p/open-ehr-gen-framework/>

⁸E-mail list, http://lists.openehr.org/pipermail/openehr-technical_lists.openehr.org/2010-November/005614.html

⁹*open*EHRApp project homepage: <http://kenai.com/projects/{\itopen}EHR-app>

	Opereffa	EHRflex	openEHR-Gen Framework	openEHRApp
Reference Model	<i>openEHR</i>	EN13606	<i>openEHR</i>	<i>openEHR</i>
Templates	not supported	not supported	supported	supported
Documentation	available	available	in Spanish	none
Kernel completeness	incomplete	incomplete	incomplete	incomplete
Service completeness	incomplete	incomplete	incomplete	incomplete

Table H.1: Comparison of Implementations

documentation on how to run the system and is more lightweight than the *openEHRApp*, which better supports testing. It was therefore chosen for the creation of the screenshots.

Using the openEHR-Gen Framework

The openEHR-Gen Framework uses their own template language. To generate screen forms with the implementation, the templates designed with the Ocean Informatics Template Designer¹⁰ needed to be converted into the template language used in the openEHR-Gen Framework. Furthermore, due to the prototype nature of the system, the archetypes used for demonstration needed to be simplified. Finally, the system is not advanced enough to support the complex and large omic archetypes.

¹⁰Ocean Informatics Template Designer, <http://www.oceaninformatics.com/Solutions/ocean-products/Clinical-Modelling/ocean-template-designer.html>

Abbreviations

ADL	Archetype Definition Language
ANSI	American National Standards Institute
AOM	Archetype Object Model
ASTM	American Society for Testing and Materials
BIMS	Biobank Information Management System
BPH	Benign prostatic hyperplasia
BTO	Brenda Tissue Ontology
CCR	Continuity of Care Record
CDA	Clinical Document Architecture
CDISC	Clinical Data Interchange Standards Consortium
CEN	Comit European de Normalisation
CV	Controlled Vocabulary
D-MIM	Domain Message Information Model
eBMRR	electronic BioMedical Research Record
EHR	Electronic Health Record
FGED	Functional Genomics Data
FuGE	Functional Genomics Experiment
FuGO	Functional Genomics Ontology
GE	Gel Electrophoresis
GelML	Gel Electrophoresis Markup Language
GO	Gene Ontology
GPIC	General purpose Information Components
GSC	Genomic Standards Consortium
GUI	Graphical User Interface
HISA	Health Informatics Service Architecture
HL7	Health Level 7
HUPO	Human Proteome Organisation
IHE	Integrating the Health Enterprise
ISA	Investigation/Study/Assay
ISAC	International Society for Advancement of Cytometry
ISO	International Organization for Standardization
LAB	Laboratory Model
MGED	Microarray Gene Expression Data
MIAPE	Minimum Information About a Proteomics Experiment

MIAME	Minimum Information About a Microarray Experiment
MIBBI	Minimum Information for Biological and Biomedical Investigations
MO	MGED Ontology
OBI	Ontology for Biomedical Investigations
OBO	Open Biological and Biomedical Ontologies
PATO	Phenotypic Quality Ontology
PCRC	Prostate Cancer Research Consortium
PIN	Prostatic Intraepithelial Neoplasia
PRIDE	Proteomics Identifications Database
PSI	Proteomics Standards Initiative
RFD	Retrieve from Data-capture
RIM	Reference Information Model
R-MIM	Refined Message Information Model
RSBI	Reporting Structure for Biological Investigations
SEND	Standard for Exchange of Non-clinical Data
spML	Sample Processing Markup Language
SOP	Standard Operating Procedure
STDM	Study Data Tabulation Model
TURP	Transurethral resection of the prostate
XDS	Cross-Enterprise Document Sharing
XML	eXtensible Markup Language

