LEABHARLANN CHOLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH | TRINITY COLLEGE LIBRARY DUBLIN
Ollscoil Átha Cliath | The University of Dublin

Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

Intuitive Human Centric Governance Of Pervasive Computing Environments

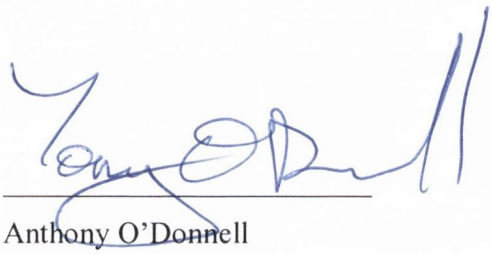
A thesis submitted to the
University of Dublin, Trinity College
for the degree of
Doctor of Philosophy

Anthony O'Donnell
Knowledge and Data Engineering Group,
Department of Computer Science,
Trinity College,
Dublin

Submitted August 2007

Declaration

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

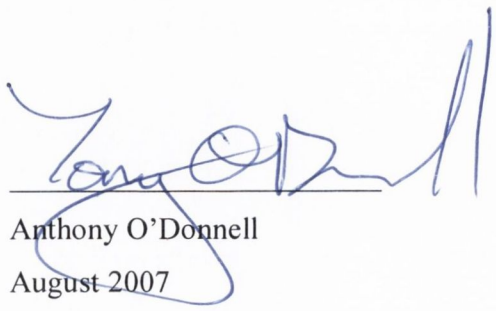


Anthony O'Donnell

August 2007

Permission to lend or copy

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



Anthony O'Donnell

August 2007

I. ACKNOWLEDGEMENTS

There a number of people I would like to thank who have helped me reach this point. Most of all I want to thank my family for all of their love, patience and support. I am absolutely blessed to have such a great family – my parents Dan and Catherine, my sister Emer and my brother Mark. They have always been willing to give me the freedom to make my own way in life, while at the same time being ready to provide support and encouragement at every step along the way. I'd also like to remember my grandparents, one who is still with me and three who live on in spirit and memory – Breade & Denis Ennis and Charlie and Susie O'Donnell, as well as Kathleen Hawkins who gave me my sense of curiosity and love of learning.

I want to say a big 'Thank You' to everyone in Trinity that helped me with this thesis, and in particular my supervisor Vincent Wade. Vinny has helped me through two theses now, and has always been a source of ideas and inspiration. I'd also like to give a special thanks to Dave Lewis for being my mentor and spiritual guide. Dave has always been a patient and available advisor and his support over the years has been invaluable.

Another two former colleagues that deserve a mention are Declan Dagger and Steffen Higel for tolerating my political rants and random musings as we shared an office. And of course, big thanks to everyone past and present in the KDEG for letting me into their club!

II. ABSTRACT

Pervasive computing proposes that in the future, human beings will be immersed in a technology rich environment, where computing power will be embedded in devices all around us. This thesis examines how such technology rich environments can be effectively governed so that they meet the demands and expectations of the users that they will support.

This examination is based around identifying and supporting three stakeholder groups. These groups reflect three different types of governance that needs to be exercised, namely: users who interact with the environment and expect support; administrators who need to model resources; and experts who can describe the types of routine behaviour that happen within the Pervasive Computing environment.

The approach to providing this governance is based around developing an integrated set of services that individually satisfy the needs of a particular stakeholder, while also interacting to provide an overall platform for user-centric governance.

Each of these services is user-centric and non-application specific. In so doing the tools cater to the needs of a variety of users, irrespective of their technical expertise or the application that the environment will support.

The services have been evaluated experimentally and each has been compared with related work. This thesis articulates the design objectives and implementations that were used to produce the experimental prototypes, and it also offers an in-depth examination of the state of the art in a number of disciplines including Pervasive Computing, Event Aggregation, Mixed Initiative, Ontologies and Autonomic Computing.

III. ABBREVIATIONS

BIF	Bayesian Interchange Format
DAML	DARPA Agent Mark-up Language
DARPA	Defence Advanced Projects Research Agency
DTD	Document Type Definition
EJB	Enterprise Java Beans
FOAF	Friend of a Friend
HCI	Human Computer Interface
J2EE	Java 2 Enterprise Edition
MPEG	Moving Pictures Expert Group
OIL	Ontology Inference Layer
OWL	Web Ontology Language
RDF	Resource Definition Framework
RFID	Radio Frequency Identification
SDK	Standard Development Kit
SOAP	Simple Object Access Protocol
SOUPA	Standard Ontology for Ubiquitous and Pervasive Applications
URI	Universal Resource Indicator
URL	Universal Resource Location
VOIP	Voice Over the Internet Protocol
W3C	World Wide Web Consortium
WSDL	Web Services Definition Language
XSD	XML Schema Definition
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

IV. GLOSSARY

Term	Definition
Cardinality	A measure of the number of members of a given set.
Description Logics	A family of languages used to represent terminological knowledge of a given domain in a structured and formally understood way.
Existential Quantification	A logical predicate applying to at least one member of a domain – typically denoted by the symbol ∃.
Governance	An attempt to control systems by describing high level goals and expectations.
Intent	A particular user goal that the system may be able to support.
Management	An attempt to control systems through highly specified and formal sets of targets and configurations.
Marginal Utility	A method of measuring of the cost or price of a choice such that it represents the least cost for the most feasible result.
Mixed Initiative	An approach to human computer interaction that seeks to provide timely and appropriate system interruptions or interventions based on a measure of the marginal utility of such actions.
Ontology	A data model representing concepts within a domain and the relationships between them.
Pervasive Computing	An emerging area with Computer Science that envisions spaces containing large numbers of embedded computing devices that seek to support users.
Universal Quantification	A logical predicate applying to all members of a domain – typically denoted by the symbol ∀.

I. ACKNOWLEDGEMENTS 5

II. ABSTRACT 7

III. ABBREVIATIONS 9

IV. GLOSSARY 11

1. INTRODUCTION 17

1.1. MOTIVATION 17
1.2. THESIS QUESTION 19
1.3. OBJECTIVES 19
1.4. DESIGN OVERVIEW 22
1.5. CONTRIBUTION TO RESEARCH 23
1.6. THESIS OVERVIEW 24

2. GOVERNANCE & USERS IN PERVASIVE COMPUTING 27

2.1. INTRODUCTION 27
2.2. WHAT IS PERVASIVE COMPUTING? 27
2.3. WHAT IS USER-CENTRIC GOVERNANCE? 28
2.4. GOVERNANCE ROLES 28
2.5. REFINING & IDENTIFYING USER GOVERNANCE ISSUES 34
2.5.1. Presentation Scenario 35
2.5.2. Presentation Analysis 35
2.5.3. Meeting Scenario 36
2.5.4. Meeting Analysis 38
2.5.5. Kitchen Scenario 39
2.5.6. Kitchen Analysis 40
2.5.7. Common Challenges Analysis 41
2.6. DESIGN RECOMMENDATIONS 41
2.6.1. Ordinary User 41
2.6.2. Environment Administrator 42
2.6.3. Domain Expert 42
2.6.4. Overall 42

3. STATE OF THE ART 43

3.1. INTRODUCTION 43
3.2. USER CENTRIC ADAPTIVITY 43
3.2.1. Event Aggregation 44
3.2.2. Bayesian Networks 46
3.2.3. Mixed Initiative Systems 50
3.3. RESOURCE REPRESENTATION & REASONING 52
3.3.1. Ontologies 53
3.3.2. Web Ontology Language – OWL 55

3.3.3. The SOUPA Ontology 57

3.3.4. Reasoning Tools 62

3.3.5. Protégé 65

3.3.6. Protégé-OWL 66

3.4. AUTONOMIC COMPUTING & GOVERNANCE OF PERVASIVE COMPUTING 67
3.5. SUMMARY 69

4. DESIGN & IMPLEMENTATION 71

4.1. INTRODUCTION 71
4.1.1. Overview 71

4.1.2. Chapter Structure 72

4.1.3. Overall Design Assumptions 73

4.2. INTENT INFERENCE ENGINE 73
4.2.1. Process Analysis 74

4.2.2. Prototype Objectives 76

4.3. ENVIRONMENT MODELLING 79
4.3.1. Process Analysis 79

4.3.2. Prototype Objectives 81

4.4. ROUTINE BEHAVIOUR DESCRIPTION TOOL [RBDT] 84
4.4.1. Process Analysis 85

4.4.2. Prototype Objectives 86

4.5. SUMMARY OF DESIGN REQUIREMENTS 88
4.6. IMPLEMENTATION 89
4.6.1. Common Technologies 89

4.7. INTENT INFERENCE ENGINE 89
4.7.1. Prototype Outline 90

4.7.2. Summary 93

4.8. ENVIRONMENT MODELLING TOOL 93
4.8.1. Limitations 101

4.8.2. Summary 102

4.9. ROUTINE BEHAVIOUR DESCRIPTION TOOL 103
4.9.1. Overview 103

4.9.2. Summary 117

4.10. OVERALL ARCHITECTURE 118

5. EVALUATION 121

5.1. OVERVIEW 121
5.2. EVALUATION OF THE INTENT INFERENCE ENGINE 121
5.2.1. Experimental Objective & Means of Evaluation 121

5.2.2. Evaluating the Objectives 122

5.2.3. Experimental Approach 123

5.2.4. *Results* 124

5.2.5. *Discussion* 125

5.3. EVALUATION OF THE ENVIRONMENT MODELLING TOOL 128

5.3.1. *Experimental Objective & Means of Evaluation* 128

5.3.2. *Design Objectives* 129

5.3.3. *Evaluation of Objectives* 130

5.3.4. *Experimental Description* 131

5.3.5. *Results* 132

5.3.6. *Discussion* 135

5.4. EVALUATION OF THE ROUTINE BEHAVIOUR DESCRIPTION TOOL 137

5.4.1. *Experimental Objective & Means of Evaluation* 137

5.4.2. *Design Requirements* 138

5.4.3. *Evaluating Design Requirements* 138

5.4.4. *Evaluation Approach* 140

5.4.5. *Results* 141

5.4.6. *Discussion* 143

5.5. CRITICAL REVIEW 145

5.5.1. *Overview* 145

5.5.2. *Integration* 146

5.5.3. *Intuitiveness* 148

5.5.4. *Returning to the Scenarios* 150

5.6. RELATED WORK 153

5.6.1. *Governance of Pervasive Computing Environments* 153

5.6.2. *Mixed Initiative Interfaces* 154

5.6.3. *Interactive Ontology Creation* 155

5.6.4. *Overall Review* 157

6. CONCLUSIONS 159

6.1. INTRODUCTION 159

6.1.1. *Overview* 159

6.2. THESIS QUESTION 159

6.3. OBJECTIVES 160

6.3.1. *Investigation of Governance Requirements* 160

6.3.2. *Investigate and Develop Prototypes* 161

6.3.3. *Perform Experiments* 162

6.4. CONTRIBUTION TO THE STATE OF THE ART 164

6.4.1. *Governance of Pervasive Computing Environments* 164

6.4.2. *Minor Contributions* 164

6.5. PUBLICATIONS 165
6.6. FUTURE WORK 167
6.6.1. Expanded environment 167
6.6.2. Integrating with pervasive computing middleware 168
6.6.3. Service Composition 168
6.6.4. User Modelling 168

APPENDIX I. REFERENCES 169
APPENDIX II. INTENT INFERENCE ENGINE 187
APPENDIX III. ENVIRONMENT CONFIGURATION TOOL 203
APPENDIX IV. RESOURCE DEFINITION FRAMEWORK (RDF) 221
APPENDIX V. AN OWL ENVIRONMENT MODEL 227

1. INTRODUCTION

1.1. MOTIVATION

"The Disappearing Computer" [Weiser, 1991] presents a vision for the future of computing where devices 'disappear' by becoming seamlessly integrated into the background environment. This approach seeks to liberate the ordinary user. It allows them to focus on their needs and expect assistance without having to learn how to make use of each device. The user is therefore free to interact with the pervasive computing environment while it adaptively responds to their needs.

This shift from an explicit flow of information from the user to a more implicit relationship with computer systems must be managed to be effective [Sloman, 2001]. Management is a concern in any complex system. In pervasive computing environments, this complexity is exacerbated by the potential proliferation and distribution of configurable devices which, taken together, constitute the system [Humble, 2003].

In addition to device complexity, such management should be enabled by tools that allow end-users of varying technical skill and experience to fully exploit available resources. This challenge arises because the expectations of such users will be set by their new relationship with pervasive computing, and by the fact that it is not feasible to expect such users to become expert at the variety and quantity of devices that will constitute the system. As such, any attempt to address management should seek to minimise the cognitive load [Sweller, 1988] placed on the user.

Management concerns in pervasive computing include a number of areas including resource level [Baccarelli, 2005], [Banerjee, 2005], [Denaro, 2005] through to responding at end-user level [Bihler, 2005] [Sousa, 2005] [Bardram, 2004] [Yau, 2004]. Typically this management is focussed on very specific demands stated in an explicit and formal way.

The examination of end-user management concerns presented here considers a more abstracted form of management where expectations and intentions drive the configuration of support. It focuses on enabling and responding to information flows between the user and the environment, rather than depending on exhaustively complete specifications. It examines how users can exercise control in a natural way [Gajos, 2002]. This thesis denotes the challenges associated with providing such management '*User-Centric Governance*'.

These flows divide into two categories: *usage flows* and *management flows*.

'Usage flows' describes the data that is gathered from sensors and context and used to make inferences about a user's intentions. These inferences can then be used to pre-emptively provide user support. For the purposes of this thesis, the collection of this data is considered beyond scope and so the examination instead focuses on how to aggregate and respond to this data.

'Management flows' represent the data that is required to ensure that the inference process results in accurate and appropriate interventions.

It has been noted that such usage and management flows are essential when responding to user activity [Abowd, 2000], and the following are key concerns:

- Developing natural interfaces to enhance the flow of information between the system and the user, and to support the more common forms of human expression such as gesture and voice.
- Ensuring that pervasive computing systems are context aware and can react to both sensed and computationally available information streams. Computational information would include any context data that is made available such as resource information, calendars or user profiles.
- Provision of mechanisms for capturing user experience such that it can be reused in the future.

In viewing these from a governance perspective, the following challenges can be identified:

- Providing a mechanism for inference which facilitates the aggregation of usage information flows
- Equipping users with a means to govern how this inference process relates input data to output conclusions
- Ensuring that context can be modelled such that it can be used to assist inference

1.2. THESIS QUESTION

This thesis assesses the extent to which human demands on the services and resources of a pervasive computing environment, and human knowledge on the routine modes of behaviour can be used to govern pervasive computing environments. This examination will focus on how human knowledge can be intuitively captured and used to programme Bayesian Network-driven, mixed initiative systems, and how sensed and context data can be used to drive the appropriate pre-emptive delivery of support to users.

1.3. OBJECTIVES

The objectives of this thesis are to:

- **Investigate the governance requirements** for the flow of information between end-users and the system
- **Investigate and develop prototypes** that demonstrate an integrated approach to such governance
- **Perform experiments** to evaluate the proposed approach and prototypes

Governing Information Flow

As noted previously, the flow of information between users and the system will be an important concern in pervasive computing systems. From a governance perspective, this flow has two channels:

- the system needs to provide an inference mechanism for interpreting the flow of information from sensors and context repositories in order to suggest support
- there is a further need to enable a flow of management information that can be used to guide inference and also to model context

Much work is being done in the area of capturing data from sensors and other sources in pervasive computing environments [Shen, 2004] [Neustaedter, 2005] [Greenberg, 2005]. The examination presented here assumes that this capture process can produce useful data on behaviour in the environment, and so instead focuses on how to analyse this data to arrive at useful conclusions on the needs of users.

As noted in 1.1, investigating the management flow will involve examining how to govern the inference process and how to model context.

Investigate & Develop Integrated Prototypes

In order to examine the inference process, its governance and the modelling of relevant context, a system based around three prototypes are proposed. While each can be viewed as a standalone component, there is a semantically and syntactically integrated flow of information between the elements so that the outputs from one can be seamlessly used as inputs to another. There is also a common approach to interface design on the management tools; this is to provide a consistent user experience.

Taking the inference process first: a mixed initiative [Horvitz, 1999] approach is proposed. Mixed initiative systems allows input data to be aggregated and for decisions to be taken based on a measure of marginal utility [Mises, 1966],

that is they take account of the relative usefulness to the user of taking action versus withholding support.

Commercial applications of mixed initiative systems include the Microsoft Office Assistant, as well as forming the basis for research in Attentive User Interfaces [Vertegall, 2003] [Horvitz, 2003]. The underlying artificial intelligence approach that enables this mixed initiative prototype is Bayesian Analysis [Moënne-Loccoz, 2003] [Horvitz, 1999].

The proposed governance method for this process is via a graphical tool that allows users to associate patterns of behaviour with high-level outcomes. In order to achieve this, a novel approach to capturing the abstract notion of utility is outlined, including an algorithm that can convert the graphical representation of inter-related events to a Bayesian Network. The above inference process can then use this network.

The third prototype addresses the need to model context such that it can assist inference, and in particular will focus on how to model devices. Such models are useful in pervasive computing applications [Held, 2002]. However, this thesis also examines whether appropriately defined models might also provide a means for configuring these devices [McGuinness, 2003].

Pervasive computing environments will contain a proliferation of devices [Mühlhaüser, 2000], and it would be preferable for these devices to be governable so that they can be tailored to the changing needs of users [Grimm, 2000]. It is therefore worthwhile to produce models that can assist such governance, as they will address the context demand defined here and also contribute to Sloman's governance challenge.

The prototype consists of an ontology-based tool that enables a meaningful dialogue to take place between the governor and the system to produce semantically complete models. This prototype also interacts with a commercial 3-dimensional map editor to capture location and dimension specific data about resources.

Perform Experiments to Evaluate Prototypes

Each of the prototypes has been evaluated to assess how well they met their design requirements, and also how they contribute to the overall goal of user-centric governance. The results of these evaluations are presented in Chapter 5.

The evaluation process involved a number of approaches including:

- Usability questionnaires, including some corresponding to ISO standards
- Quantitative analysis of how test subjects performed at completing specified tasks by examining their completed work
- Observation of subjects to provide qualitative feedback on how they interacted with the tools
- Recorded commentary from each of the subjects to give an insight to their thought processes
- Examination of the various outcomes and artefacts produced by the experiments

1.4. DESIGN OVERVIEW

The proposed system consists of three components: an Intent Inference Engine; an Environment Modelling Tool; and a Routine Behaviour Description Tool. The tools are used to provide configuration and context data for the inference engine, which in turn responds to user activity to deliver useful and timely support.

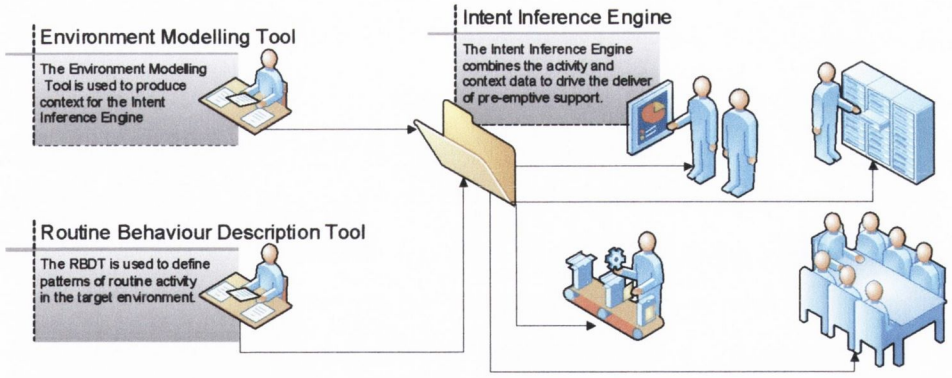


Figure 1.2 – The Integrated Governance System in Practice

In Chapter 4, the design and implementation of the prototypes for each of these components is presented in detail starting with the Intent Inference Engine. This component was the first one to be developed, and it was used to drive the requirements for the two configuration tools. As the engine will be the point at which user support is selected, it will be the component at which the data produced by the management flows and usage flows is combined to produce inferred outcomes.

As this is the order in which the components were identified, this is the order in which they are discussed in Chapter 4 and evaluated in Chapter 5. However, in practice, the operation would be as shown in Figure 1.2 with the Environment Modelling Tool and RBDT being used to prepare the engine in advance of it being required to perform inference.

Each of these components addresses a different type of governance, and each exists as a separate element. However, they are linked via the demands of the inference engine which sets out a clear interface and specific demands on the semantics and format of data produced by the other components.

1.5. CONTRIBUTION TO RESEARCH

The principle contribution made by this research is that it demonstrates a means of supporting user centric governance that is non-application specific. This is achieved by using a novel approach that employs an integrated set of

services; collectively these services support user centric governance for a broad set of business concerns. The services themselves are agnostic about the types of activity that the target pervasive environment will support, and instead they are focussed on the functional requirements of general user centric governance such as inferring outcomes based on observed behaviour, modelling context and describing routine behaviour in terms of the user's goals.

The approach specifically contributes to a number of areas within pervasive computing. The system demonstrates how the mixed initiative approach currently employed in desktop computing can be exploited to deliver intent inference in pervasive computing environments.

Furthermore, it assesses whether users with little or no prior knowledge of the system can express their valuable knowledge on routine behaviour, as well as define their demands on devices. This approach empowers users by allowing them to directly and intuitively govern the inference process.

As noted a generic approach is used, which allows users with specific knowledge to tailor the system for their application. This contrasts with other pervasive computing research, where the system itself is designed to address domain-specific needs such as supporting shoppers [Bohnenberger 2005] [Wasinger, 2005], using a mirror as a personalised display [Fujinami, 2005] and using pervasive environments to monitor for health problems [Morris, 2005].

In addition to contributing to pervasive computing, the thesis also proposes a novel use of ontologies to facilitate a user-system dialogue. This contrasts with the conventional application of ontologies as a framework to facilitate inter- and intra-system sharing of knowledge.

1.6. THESIS OVERVIEW

This thesis investigates an integrated approach to user-centric governance of pervasive computing environments. The approach is focussed on providing a

mechanism for responding to implicit user demands presented to the system by sensors and context data; it then goes on to examine how to govern this system to ensure the appropriateness of support and also to allow users of limited system experience to fully exploit system resources.

The investigation begins with an examination of user-centric governance. This includes an analysis of some potential pervasive computing scenarios and a critical examination of the challenges they pose with regard to governance. The chosen scenarios cover a range of activities, which illustrate issues for home users, business users, individuals and collaborative users. The analysis demonstrates how certain concerns are common across pervasive computing activities, while others are particular to a particular set of tasks or a particular application domain.

In addition, this analysis identifies three user roles that mirror the need to respond to user activity, govern the inference process and model context. Once defined, these roles are used to perform experiments on the proposed prototypes.

Following this, a broader examination of the state of the art in related fields is offered. This examination particularly focuses on user-centric adaptivity, resource and activity representation and the use of autonomic computing approaches in pervasive computing. It involves an analysis based on ease of use, ease of knowledge capture, availability of tools and technologies and examples of their use in pervasive computing.

Based on the roles identified and the technologies surveyed, the thesis then moves on to describe a potential design that draws on these examinations and seeks to meet the overall objectives of the thesis. This is followed by a discussion about the prototype implementations. These prototypes are then experimentally evaluated and the results are presented.

The thesis concludes by critically examining the experimental results with regard to the initial objectives of the thesis. These conclusions identify the

strengths and weaknesses in the proposed designs, their contribution to the state of the art and some further work that may be pursued.

2. GOVERNANCE & USERS IN PERVASIVE COMPUTING

2.1. INTRODUCTION

In Chapter 1, the notion of user-centric governance in pervasive computing was briefly introduced, as were some of the challenges associated with realising it. This chapter expands on these themes and demonstrates the importance of user-centric governance. It also identifies the design requirements for such a governance system.

In particular it begins by revisiting how this thesis views pervasive computing and its governance. It then identifies a number of roles, which match the need to respond to user behaviour, govern the system's inference mechanism and model devices. It then discusses a number of scenarios that illustrate different types of pervasive computing environments, and offers a critical analysis of the governance challenges they present.

2.2. WHAT IS PERVASIVE COMPUTING?

In laying out his vision for pervasive computing, Weiser describes it as "the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user" [Weiser, 1991]. He goes on to note that such 'disappearing' technologies are often the most profound.

Pervasive computing represents an evolutionary step in the human computer relationship. This relationship has already developed through the early mainframe model of multiple users on a single machine, to the PC age of a single machine and a single user, then on to the consumer electronics age where individual users have multiple devices. Pervasive computing represents a many-to-many relationship, where prolific computing resources are embedded in the environment and made available to duly entitled users.

At a user-level, one of the key features of pervasive computing environments is that the flow of information between humans and the system is often implicit [Taffat- Bouzid, 2005]. In these environments, users interact through natural interfaces, and the system responds to these sensed inputs to provide appropriate, pre-emptive support. This response process is enhanced by the incorporation of context information to deepen the system's understanding of the user's intentions [Wu, 2002].

2.3. WHAT IS USER-CENTRIC GOVERNANCE?

As noted above, the ability to respond to sensed and context data is a powerful feature in pervasive computing environments. To be successful, such environments need to be capable of meaningfully combining these input data to enable inferences to be made regarding the intentions and needs of users. However to be truly user-centric, users must also be able to exercise control in a natural way [Gajos, 2002]. This thesis denotes the challenges associated with providing such management over the inference process 'User-Centric Governance'.

2.4. GOVERNANCE ROLES

In order to explore the demands of user-centric governance, it is worth examining the types of users who will either benefit from effective governance or exercise governance themselves. In Chapter 1, some key challenges are outlined which this thesis examines, namely: responding to sensed and context input data; governing the inference process; and modelling devices. Each of these three concerns can be bound to issues faced by three different user roles. These three user roles are outline in Table 2.1.

These roles derive from the types of governance outlined in Chapter 1 namely implicitly demanding support through interacting with the environment (Ordinary User), modelling context to guide inference (Environment Administrator) and governing the inference process (Domain Expert). While these are by no means an exhaustive set of potential stakeholders, they do

represent three different types of governance: implicit control of the environment itself; configuring the mechanism underlying the delivery of support; and modelling context to inform inference.

Role	Governance Need
Ordinary User	Interacts with the environment and needs to receive appropriate support
Environment Administrator	Needs to model the devices based on the demands they want to exercise
Domain Expert	Has knowledge of routine behaviour that needs to be captured to govern inference

Table 2.1 – Governance Roles

Each of these governance roles is described below.

It is worth noting that while each role has been identified as a separate concern, this does not mean that the same user cannot fulfil more than one role simultaneously. This is examined in detail through the scenarios presented later in this chapter.

Ordinary Users

As has been seen, pervasive computing has been promoted as a way to improve the relationship between human beings and computers. These 'Ordinary Users' are the people who will make use of the pervasive computing environment and will have individual expectations about how the environment should meet their changing needs. The purpose of the governance system is to ensure that this support is appropriate and acceptable to the user.

A pervasive computing environment should enable the following:

- Support user interaction through natural interfaces
- Support should take account of sensed and computational inputs
- The system should capture user experience to assist future support

As noted in Chapter 1, Abowd's first two requirements address managing interaction and sourcing data streams [Abowd, 2000]. This is an active area of research in pervasive computing environments [Shen, 2004] [Neustaedter, 2005] [Greenberg, 2005]. This thesis will not examine sourcing sensor data as it assumed that such data will be available from elsewhere [Garofolakis, 2006] [Jeffery, 2006], and is therefore beyond the scope examined here. This layered separation of sourcing data and performing inference on it has been demonstrated as an effective way of contributing similar systems [Oliver, 2005]. The challenge therefore becomes one of interpreting and responding to such data, such that this interpretation process is governable.

The third point suggests a need to provide a means for capturing user experience as a way to enable feedback. This is not examined in this thesis; as such an examination could shift the focus onto artificial intelligence and machine learning and away from user level concerns identified thus far.

The research objective is therefore to provide a system that facilitates the other requirements, and can be governed in order to deliver appropriate support. By providing such a system, specific governance demands can be identified and solutions to them can be evaluated experimentally.

As stated in the thesis question, and elsewhere in Chapter 1, the underlying approach to supporting users examined here is based on a mixed initiative approach to aggregating input data and inferring appropriate pre-emptive interventions by the system on the user's behalf.

The following are general challenges in developing mixed initiative systems:

- Provide a mechanism for aggregating input data
- Produce utility-driven requests for support

In governance terms, the challenge is to configure the mechanism for aggregating data so it can be used to deliver support.

Environment Administrators

The environment administrator is responsible for modelling devices in the target pervasive computing environment. These models fulfil two roles: acting as a context input to the inference process; and potentially providing configuration data for devices in the space. The proposed approach considers how an appropriate configuration method can be used to produce models that fulfil both requirements.

There are a range of potential approaches to configuring devices in a pervasive computing environment including: an entirely automatic configuration [Connelly, 2003]; the development of a common configuration interface available on all devices [Zeidler, 2001]; or individually configuring devices using their native interface as routinely happens with current user devices, e.g. iPod, mobile telephone, VCR, games console, PC etc.

In environments where there is heterogeneity of devices, such approaches may not scale well or offer the level of adaptivity and intuitiveness that is sought. This scalability problem is due to the potential increase in administrative overhead and cognitive load [Sweller, 1988] placed on users if they must individually configure devices, while a common interface will be limited by the capabilities of the least well-resourced device [Zeidler, 2001]. It is also arguable as to whether an entirely automatic system involves any level of user-centric governance, as it will be bound by whatever drives its automation.

In order to model devices for a particular task, three specific requirements have been identified:

- Intuitively capture the administrator's requirements
- Identify a mechanism for converting these requirements into a configuration model
- Ensure that the resulting configuration is sufficient for both the context demands of inference and potential device configuration

The approach examined here adopts some of the autonomic computing philosophy [Kephart, 2003], which proposes managing complex systems by means of high-level policies that are interpreted by the component sub-systems. In so doing, this thesis argues that it provides some automation without being entirely automatic, while also allowing governance to be performed at a high-level without a need to configure individual resources.

The proposed method for intuitively capturing the administrator's requirements is based on the use of ontologies to drive a dialogue between the system and the user to produce semantically complete models. The component also interacts with a 3D map editor which allows users to view their target space and describe the location or dimensions of resources in an accessible way.

Domain Experts

In Abowd's requirements for pervasive computing environments, there is a need to enable such environments to respond to sensed and context inputs. In the user support platform described here, a utility-based approach to aggregating input data is proposed. The provision of such utility-based models has been identified as a necessary step in empowering users in human-centric pervasive computing systems [Gajos, 2002].

This aggregation process needs to be governed in order to ensure that the responses are appropriate. To enable these appropriate responses, this thesis investigates if aggregation can be driven by existing human knowledge of routine behaviour in the target environment. In particular, the examination

focuses on how knowledge about routine activities and their intended outcomes can be used to carry out inference on observed behaviour in order to deliver support. The governance challenge being explored is therefore limited to capturing this knowledge in such a way that it can be used to deliver utility-based inference. The source of this knowledge is termed a 'Domain Expert'.

These domain experts will know the types of behaviour that underlie routine activities. It is therefore necessary to examine methods for capturing this knowledge so that activities can be related to potential outcomes.

To address the need to capture this expert knowledge, and taking account of the over-arching goal of intuitiveness, a number of requirements have been identified:

- Provide a tool for capturing behavioural knowledge
- Provide a method for incorporating utility into this capture process with a minimum of cognitive load
- Ensure that the behaviour models can be used by the user support platform

To investigate this, a behaviour capture tool is proposed taking the form of a graphical application. This tool is examined to see if it allows the domain expert to associate basic activities together to form more complex patterns of behaviour. This is done via a network of nodes and directed edges representing activities and the causal links between them. This evaluation assesses whether it allows observed and context data to be interrelated in order to facilitate future inference.

The tool also proposes a novel approach to capturing utility which involves the domain expert indicating the relative influence of activities on each other by considering the relative distances of the nodes representing them, and by setting a threshold on each edge to represent the level of confidence needed for it to be assumed true. The experimental assessment examines whether users can express the abstract notion of utility in a way that minimises cognitive load. This involves a comparative experiment where users attempt similar tasks

using the proposed tool and also through an established Java-based Bayesian Network editor called JavaBayes [Cozman, 2000].

This resulting utility-based network can be tested by the user support platform to examine if it can deliver mixed initiative support to users. Bayesian Networks have already been shown as an effective tool in mixed initiative systems, and so a means of converting the utility-based network to a Bayesian Network is also examined.

In adopting this approach, this thesis seeks to provide a feasible, integrated governance solution to an Abowd-type pervasive computing environment.

2.5. REFINING & IDENTIFYING USER GOVERNANCE ISSUES

In order to further examine the demands imposed by user-centric governance, a number of scenarios are presented and analysed below. These scenarios represent a variety of application domains and types of user activity such as home and business settings, and individual and collaborative activities. After each scenario, an analysis of their governance issues is presented which focuses on the particular affordances associated with that scenario and how it impacts on the three roles discussed above. A review of common governance issues is presented at the end.

The scenarios have been chosen as exemplars of the different kinds of relationships that a pervasive computing space has with the users within it. The *Presentation Scenario* features a situation where there is one lead user and a number of somewhat anonymous participants; the *Meeting Scenario* involves a gathering of peers competing for system resources; and the *Kitchen Scenario* examines a highly personalised relationship between a homeowner and their system.

2.5.1. Presentation Scenario

It is the end of term and the second year Economics class have just completed their term group project. Each group must present their results to the class at a session where their classmates can challenge their report's findings.

The class enters the lecture theatre, with the majority taking their seats, while the presenters make their way down to the front. Sheila, the course lecturer, has already decided the order of speakers and alerted the system to this order. As a result, the slides for the first group are loaded when their Presenter makes their way to the lectern.

From time to time, members of the audience raise their hand to ask questions. The presenter signals that they will take the question by pointing at the student concerned. Once a question has been allowed, a directional microphone directs itself at the questioner, and their voice is replayed over the speaker system and added to the recording of the session.

When a speaker approaches the end of their allotted time, the system discreetly alerts them via the lectern terminal. When the time has fully elapsed the lecturer can allow additional time with a suitable phrase. Failing that, the lectern microphone is disconnected and the student makes their way back to their seat.

At the end of the session, the presentations are placed online and the class is forwarded a link to their location as well as that of the recording of the session.

2.5.2. Presentation Analysis

Specific Affordances

This scenario represents a situation where much of the activity in the space does not result in any support being offered. For example, any chatting or casual interaction between members of the audience does not result in any

system response. The scenario is instead focussed on a strong notion of roles (lecturer, presenter, audience member) and the nature of the inference that takes place is strongly influenced by these. It would therefore be necessary for some context service to operate in this scenario that could monitor the occupiers of the various roles, especially as they change during the course of the session with audience members becoming presenters and back again.

Impact on Roles

In this example the lecturer would most likely fulfil all three governance roles while the students fall into the category of ordinary users. As lecturer, Sheila is responsible for ensuring that the facilities are adequate for the presentation scenario. However, in a large enterprise such as a university where presentations are routine, and the lecture theatres are specifically intended for such activities, it is likely that she would be able to reuse existing configurations and models for this exercise rather than having to construct them herself. The governance challenge therefore becomes one of selecting which of the potential outcomes she wants to enable (such as recording the session), and also defining access privileges to resources.

2.5.3. Meeting Scenario

It is Tuesday afternoon and the Systems Research Group arrives into the seminar room for their weekly meeting. John, the group's leader, has already logged into the room management system in order to configure it for today's meeting. This included equipping the room's phone system with international dialling capabilities for this afternoon's conference call, ensuring that a live stream be delivered to Peter as he will be attending remotely and advising the room's management system of the group members. He also requests that the meeting be recorded.

The group members enter the room and John makes his way to his usual seat at the top of the table. The other members take seats around the meeting table. Once everyone is in place the meeting gets underway with a review of the

previous week's action plan. The system displays a copy of the previous week's plan on each terminal, and as points on it are dealt with, it updates the plan in order to forward an update to the group's inboxes after the meeting.

The main parts of today's meeting are a presentation from Sandra and a discussion of revisions of the current prototype the group are working on. John calls on Sandra and she makes her way up to the lectern. The system loads Sandra's presentation from her file storage and when she indicates she's ready to begin, the lights dim, the presentation's first slide appears and the recording system focuses on Sandra and stops recording interventions from other attendees.

Sandra begins her presentation; halfway through the fourth slide Peter raises his hand. Sandra pauses to allow the question. The system, recognising that she is allowing the question, begins to record Peter. After dealing with the question, Sandra proceeds with her presentation, answering some further questions. When she completes the final slide the lights come back on. She invites further questions, and a question and answer session begins.

About ten minutes later, John realises that the allotted time has expired and interrupts to bring Sandra's presentation to an end. The system acknowledges John's role and allows this interruption. The meeting now moves onto a discussion of the group's current prototype.

Peter is the lead designer of the prototype, and he asks for an image of the current design to be displayed on the whiteboard. The same image is relayed to Peter's terminal. He gives a brief report of the prototype's progress.

After he finishes, Anne walks up to the whiteboard and suggests some design changes. Sandra points out flaws in two of them, but the others seem like possibilities. A discussion of these then ensues and after some more minor alterations, an updated design is saved from the whiteboard.

The last main piece of business is a conference call with the group's partner in Amsterdam. Having previously equipped the room with international dialling

capabilities, John puts the call through and the system monitors the group for contributions. At the end of the call it is agreed that a further call will be necessary. John asks the diary service to suggest some suitable times, which it does by examining the members' diaries. Once a suitable time is agreed the call ends, as does the meeting.

The system places a copy of the recording of the meeting online and emails the attendees with a link to the recording.

2.5.4. Meeting Analysis

Specific Affordances

This scenario reflects a great deal of collaborative interaction, and a need to deliver simultaneous support to all of the users by monitoring all of their activities. This contrasts with the presentation example, which reacted only to a limited set of inputs and where the real focus was on one actor at a time (the presenter) with system actions occurring at a group rather than an individual level.

Another difference in this scenario is the need to track user roles over the course of the session is not as important as the presentation scenario. Indeed, roles in and of themselves are not a key concern, as unlike the university example, all of the users are effectively peers and have equal access to resources.

Another specific concern in this example is that not all actors are in the same location. This would require some bridging between the different locations to ensure that the system's approach to support was consistent.

Impact on Roles

John acts as the environment administrator as he configures the room for the meeting. As with the presentation example it is likely that John would also

perform some of the domain expert duties, however, as with any large corporation, it is likely that much of the standard operating procedures that define the protocols for meetings such as this are set centrally and so as with the presentation it would be John's duty to tailor them for the particular needs of his meetings, such as his requirement that it be recorded.

2.5.5. Kitchen Scenario

Mark wants to prepare a meal. He decides that he wants to make lasagne. He has not made lasagne before, but he has found a recipe on the Internet. All of the ingredients are stored in his kitchen in different cupboards and the refrigerator.

To begin, Mark requests that the recipe be displayed. There is a terminal next to the counter he is working at and the recipe appears on the display. He then takes out the ingredients for his sauce starting with tinned tomatoes and some herbs. The recipe recommends he put these over a low heat so having placed them in the pan; he puts it on the hob. Initially he puts this on a too high a temperature. When he fails to lower the temperature, the hob discreetly attracts his attention and so he turns the temperature down.

Mark continues to prepare his vegetables. The recipe suggests that these first be fried in olive oil. He begins to search for the oil, but he can't remember where he last stored it. As he looks for the oil, the LED on the door of the cupboard in question begins to blink but Mark doesn't notice. After another few moments, the system projects a direction to the correct cupboard onto the counter next to him. Having taken note of the direction, Mark goes to the press and retrieves the oil.

Cooking proceeds, and once the meat and the sauce is prepared; Mark checks the recipe again and sees that he needs to make a white sauce. The recipe doesn't give the ingredients for this and so Mark pauses for a moment. His mother makes a good lasagne, and so he decides to contact her for advice. He verbally asks to call his mother and the system puts him through to her using a

local VOIP service. Having received directions from her, including a recipe for white sauce, he finishes off the lasagne and puts it in the oven for thirty minutes.

He then relaxes with a cup of coffee and starts to read a magazine. After thirty-one minutes, Mark still hasn't removed the lasagne, and so the system again attracts his attention. He removes the lasagne and sits down to a tasty meal.

2.5.6. Kitchen Analysis

Specific Affordances

This example differs from the others as it occurs within a home environment. In such an example it is unlikely that someone would invest considerable time and effort in explicit configuration, and is more likely that a system like this would rely on some form of self-learning and improvement over time to increasingly tailor itself to the very limited set of users it would need to support.

For example, the fact that it interrupts aggressively to point Mark to the oil might represent an overly assertive intrusion, especially if Mark was allergic to oil or normally deviated from that part of the recipe.

Impact on Roles

In such an environment it is likely that the users would rarely exercise roles other than ordinary user. These lightweight environments place an increased burden on the system to meet the governance deficit through machine learning techniques. The device model of the room is unlikely to change over time, and neither are the demands that the users will place on them. Similarly, the set of activities that will be supported are likely to be similarly unchanging. Finally, it is likely that the same limited number of users will interact with the space over time. It might therefore be sufficient in such environments for the domain expert role to be replaced by a comprehensive set of behaviour descriptions

uploaded at install time. These could adapt automatically over time based on the system's observation of activity. Similarly, domestic devices could come equipped with their own appropriate, bundled model of their capabilities that could be advertised to the system when they are first installed.

2.5.7. Common Challenges Analysis

The common challenges revolve around getting appropriate models of behaviour and devices into the system. These are essential for the delivery of support. However, as has been discussed preparing these models can depend on the nature of the environment in which such activity is to be supported. In large enterprises, which have a highly centralised approach to defining corporate policies there may be little flexibility in the way that end users can exercise governance. This could mean that governance is limited to selecting from pre-defined support options. In more delegated situation, such as a university, the level of governance might be enhanced, while in a home environment user-governance might be replaced by a 'system knows best approach'.

2.6. DESIGN RECOMMENDATIONS

By reflecting on each of these different applications and their variety of human-computer relationships, a number of design recommendations can be drawn. These recommendations are presented under the three roles identified in this chapter.

2.6.1. Ordinary User

The ordinary user needs to be supported by a suitably configured inference process. It has been noted that it may be beneficial to incorporate some type of machine learning to improve the quality of inference, especially when other governance may not be exercised such as in home applications.

2.6.2. Environment Administrator

The environment administrator needs to be provided with easy to use tools that allow her to define models that are sufficient for both the inference process and for device configuration. The definition of sufficiency will depend on the demands of these applications, and so the system that supports administration will need to be able to adapt to these demands on an application-by-application basis.

2.6.3. Domain Expert

The domain expert needs to be provided with tools that are flexible enough to allow him to exercise whatever level of governance is permitted by his enterprise. This can be achieved by separating the process by which behaviour is described from the data available for descriptions. Such an approach would allow corporate policies on devolved authority to be enforced by limiting the scope of the domain expert to govern concerns beyond their authority.

2.6.4. Overall

The system should provide a means for information to flow internally between the governance tools and the inference process, and also between the various users and the system in general.

The evaluation presented in Chapter 5 will present a further brief analysis of how each scenario might be satisfied by the proposed system design.

3. STATE OF THE ART

3.1. INTRODUCTION

Chapter 2 provides a deeper understanding of user centric governance as viewed by this thesis. This chapter presents an examination of technologies that may support the delivery of such governance. In particular it will focus on user centric adaptivity, resource representation & reasoning and the use of autonomic computing techniques in governing pervasive computing environments. The examination will include an overview of each of these areas as well as critical review of some related technologies. This critical review will hinge on the usability and availability of tools, as well as specific demands such as the ease of knowledge capture.

The critical examination will identify the technologies that are suited to the governance demands outlined in Chapter 2, and will be used to drive the design and implementation of prototypes proposed in Chapter 4.

3.2. USER CENTRIC ADAPTIVITY

User centric adaptivity refers to the ability of systems to respond to users in a tailored, personalised and dynamic fashion. This type of adaptive behaviour has been used in areas such as adaptive hypermedia in eLearning [Dagger, 2004] [DeBra, 1999] [Brusilovsky, 1998], the integration of scientific data sources [Kambhampati, 2005] and eCommerce sites [Alpert, 2003]. It is also becoming an active area in pervasive computing research where the ability to adapt to the varying needs of users is an important concern [Lewis, 2004] [Gajos, 2002].

In general, adaptive behaviour involves identifying some model of the user that will guide the choice of tailored support. This model may focus on a particular axis of adaptivity or it may be a blend of a range of factors. For adaptivity in

pervasive computing environments, these axes can be divided into behavioural models and context models [Higel, 2003].

In order to respond to these input axes, the environment needs a mechanism for aggregating these heterogeneous input streams and a method for selecting appropriate support based on this analysis [O'Donnell, 2005]. A number of approaches can be used to perform this aggregation and response such as Event Aggregation, Bayesian Networks and Mixed Initiative Systems. These are examined below.

3.2.1. Event Aggregation

High-level activities such as making telephone calls or organising meetings are composed of a sequence of much smaller actions performed by the user. Event Aggregation [Luckham, 2002] considers ways of using observed data about a user's activities to construct higher-level intentions.

The notion of event processing has been around for some time in such diverse areas as simulating faults in telecommunications networks [Schuster, 1999] and national security [Jakobson, 2004]. Projects such as the Complex Event Processing initiative at Stanford University are now trying to apply these techniques to user activities observed via heterogeneous inputs. This approach is very relevant to the pervasive computing world of heterogeneous input streams that need to be combined in order to get an overall view of activity in the environment. The CEP initiative is currently broadening its range of applications and is now proposing to reapply their work in event processing to areas such as Autonomic Computing, Business Project Management, Event Driven Architectures and IT Security.

Event aggregation has drawn on artificial intelligence techniques such as Case Based Reasoning [Jakobson, 2004] and Bayesian Analysis [Moënne-Loccoz, 2003]. The technique chosen tends to reflect the nature of the problem being considered and the type of available data.

An example might be a battlefield where behaviour is governed by military procedures [Jakobson, 2004]. These procedures translate neatly into cases for a case-based reasoning system.

Alternatively a system might examine human expressions as they change over time as a data source for inferences [Moënne-Loccoz, 2003]. Bayesian Analysis allows for causal links between observed changes to be expressed and used to recognize the sentiments being expressed in such scenarios.

When considering which AI technique best suits the demands of aggregation in pervasive computing, an important factor is the potential ambiguity associated with implicit inputs. This may make the process of capturing suitably complete and useful cases difficult. In the work discussed in this thesis, Bayesian Analysis was therefore chosen as the inference technique because as with expression recognition [Moënne-Loccoz, 2003], the challenge here will be to express human activities in terms of causally related sub-activities.

It has been demonstrated in the projects referred to above that event aggregation allows for the use of existing models of behaviour to infer potential outcomes based on observing current events. This is particularly relevant to the work being presented here.

Some technologies have been developed to deal specifically with the demands of event processing such as RAPIDE [Luckham, 1996]. RAPIDE is an Executable Architecture Definition Language [Garlan, 2000]. It is designed to support component-based development of large, multi-language systems. The framework RAPIDE uses is based on an event-based execution model of distributed, time-sensitive systems. This allows RAPIDE to create frameworks based on causally related distributed events.

RAPIDE does have some limitations however. It was evaluated as part of this thesis, and at the time did not have rich tool support or an extensive publications record, and so other approaches were adopted. However, it has since seen a research renaissance and is now a key part of the broadened CEP project.

3.2.2. Bayesian Networks

Bayesian Networks are used as a powerful analysis tool in decision-theoretic planning [Blythe, 1999] where the probability of a set of outcomes, and some knowledge about the state of a system can be used to make inferences. The networks consist of a directed acyclic graph of nodes representing variables and arcs representing the relationships between them. Nodes can have any value, but typically they may represent simple states such as true and false. When the value of a node is known it is said to be an evidence node. This information can be used to deduce the probable states of other attached nodes.

Bayesian Networks are used in event processing and mixed initiative systems, and they provide an effective mechanism for tracking multiple potential outcomes over a period of time [Oliver, 2005]. Bayesian Networks also provide a neat solution to the network-based approach to interpreting routine patterns of behaviour as sets of causally inter-related actions, as set out in Chapter 2. They are also supported by the open source JavaBayes [Cozman, 2000] application developed at Carnegie Mellon University and now supported by Hewlett Packard. JavaBayes is discussed in detail below.

Hidden Markov Models (HMM) [Rabiner, 1989] are another AI technique that can be used to make deductions based on observed behaviour. HMMs are used to reason about unknown parameters in a system by observing known parameters. These models can then be reused to drive pattern recognition applications.

They have been applied extensively in areas such as speech recognition [Rabiner 1989] [Rosti, 2002] [Jiang, 2006]. They can also be used as a means of interpreting behaviour within semantically-rich systems. Such systems already have considerable knowledge about the domain in which that behaviour occurs, e.g. interpreting gestures when it is known that they match sign language symbols [Galata, 2001].

In pervasive computing environments, research has shown that HMMs may provide a useful mechanism for interpreting sensor data; however they are seen as a complement to Bayesian Networks [Oliver, 2005]. In such systems, a layered approach to inference is adopted where HMMs are used to provide processed input data to the Bayesian Networks. They then use this data to make inferences about a user's higher-level intentions.

Case Based Reasoning (CBR) provides another alternative when considering how to aggregate behaviour data to produce deductions about likely outcomes. Typically CBR provides the best results when it is possible to define thorough cases covering all potentially interesting outcomes.

Military simulations provide a good example of this, where battlefield behaviour can be well-defined through examination of military procedures and objectives [Glinton, 2004]. This allows CBR to be used to predict outcomes by observing behaviour. However, the typical pervasive computing environments are far more ad-hoc and loosely defined than military engagements, and so the value of CBR in broader situations is limited.

Having examined the relative merits of Bayesian Networks, Hidden Markov Models and Case Base Reasoning, it was decided to implement a Bayesian Networks based solution. Bayesian Networks have been demonstrated as a useful solution to aggregating behaviour on a mixed initiative basis [Horvitz, 1998] [Oliver, 2005]. They are also supported by open source tools such as JavaBayes which is described below.

HMMs may also provide a complement to the system proposed here by providing a mechanism for applying some semantic processing to raw sensor data, so that the inference engine can be provided with useful input data.

JavaBayes

The JavaBayes application has been used in modelling user behaviour [Cohen, 2004], and while it exists a standalone application it offers an API that allows

for reuse of its functionality. An example of a JavaBayes designed network is shown in Figure 3.1.

The standalone application consists of two terminal windows. One allows users to graphically create networks and edit their properties as shown in Figure 3.1. The other presents the results of inferences carried out on the network.

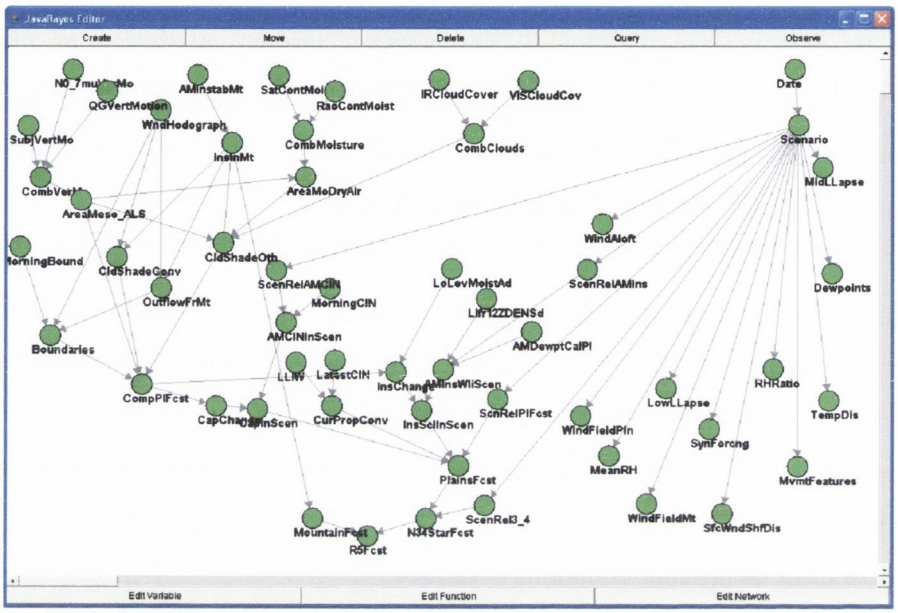


Figure 3.1 – A Bayesian Network describing the weather conditions required for hail

Bayesian Networks do pose some challenges to the system designer. In particular the creation of the networks can prove complex as not only does the designer have to create the various nodes and the links between them, they also have to define the output conditions at each node for all of the various potential input combinations from antecedent nodes. If each node has *m* states, then the number of potential output states is m^n where n is the number of input nodes.

The network shown in Figure 3.1 provides a method for forecasting hail. Consider a single node in this network, say 'PlainsFcst' which handles predictions of hail on the plains. This node has four inputs and one output. According to the network designer this node has three output states *'NIL'*, *'SIG'* and *'SVR'*, for no hail, significant hail and severe hail respectively. Its

input nodes have three, four and eleven potential input states reflecting different meteorological evidence. This leaves dozens of potential output possibilities for 'PlainsFcst', each of which have to be entered by hand with some knowledge of the significance of each state.

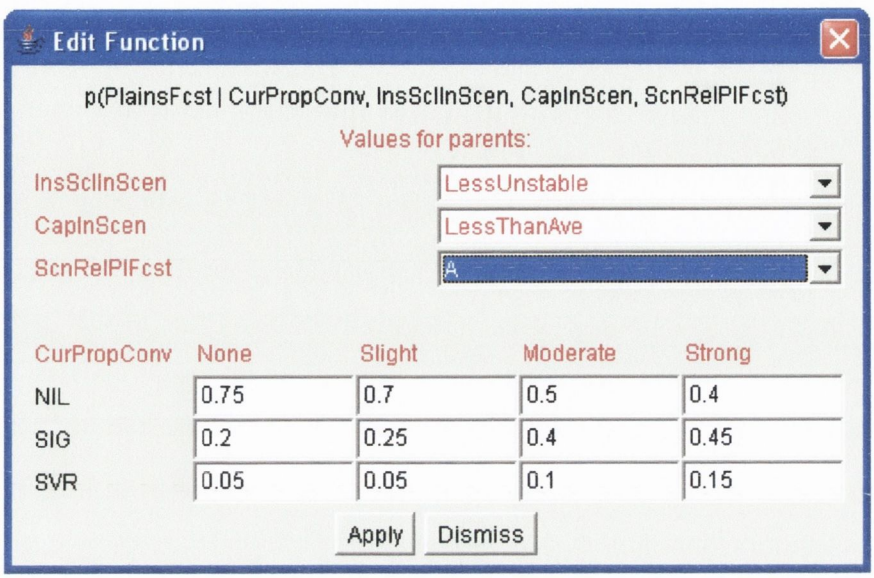


Figure 3.2 – The output states of node 'PlainsFcst'

Figure 3.2 shows the JavaBayes window for setting these output states. The user must cycle through the potential combinations of input states and expertly and precisely combine these to give output probabilities of 'PlainsFcst'.

Research has shown that a significant cognitive load is placed on users creating conditional probability tables for Bayesian Networks [Das, 2004]. For this reason a novel approach for inputting data into a Bayesian Network is proposed in Chapter 4 and 5. It abstracts the requirement of setting each individual output state and instead allows users to define the relative importance of parent nodes via their distance from the node being examined, while also allowing thresholds to be set on these input nodes. An algorithmic approach is then proposed which allows this information to be compiled into all of the potential output states.

3.2.3. Mixed Initiative Systems

Computing systems are becoming increasingly complex, while their interfaces are becoming more accessible to non-expert users. Think of your desktop computer. It is orders of magnitude more complex than the earliest electronic computers, yet its interface is simple and versatile so that users can execute a variety of tasks without any in-depth system knowledge and with a minimum of cognitive load.

In this way more and more of the responsibility for controlling the system is passing from the human user to the system itself. This is true in such diverse applications as anti-lock braking systems through to automatic updates for a computer's operating system. When these automatic decisions start to interfere with the user's normal activities, perhaps by taking inappropriate or unwanted actions, they run the risk of being disabled and thus rendered useless.

Mixed Initiative Systems [Horvitz, 1999], or as they are sometimes called Attentive User Interfaces [Vertegall, 2003], attempt to overcome this limitation by introducing the notion of utility as a means of governing when it is appropriate for the system to pre-emptively take action.

Utility is a term used in many academic disciplines, but for the purposes of this thesis it is defined as a measure of the *desirability* or *appropriateness* of taking a particular action versus not taking it. 'Desirability' and 'appropriateness' are dependent on the user, their context and the significance of the action being considered.

Mixed Initiative Systems were first proposed by the Microsoft Lumière Project [Horvitz, 1998]. This project was examining ways of offering dynamic user-support in desktop applications by attempting to infer a user's intentions by observing their activity. The project ultimately led to the 'Office Assistant' feature in Microsoft's Office programme.

Essentially mixed initiative involves building up a view of a user's activities over a given time, and based on this activity, inferring the user's intended

needs. The point at which support is offered to help reach this outcome depends on a number of factors including the nature of the task and the preferences of the user. These factors combine to provide a utility threshold that controls the point at which support is offered.

Utility is measured against four potential outcomes: support is offered correctly; support is offered incorrectly; no support is offered; no support is offered when it should have been. If the two negative outcomes, support is offered or withheld incorrectly, can be posed against the positive outcomes then one can arrive at a threshold value at which there is a greater marginal cost in offering the service and being incorrect then in withholding it. This is the governing principle of mixed initiative systems. When an appropriate level of confidence is reached support can then be offered.

This is illustrated in Figure 3.3 where the left area denotes conditions where no support is offered given the competing values of utility 'u' computed from the observed action A and the inferred goal G. The probability of offering support to achieve G given the observed evidence E is P (G|E) and when the crossover threshold is reached at the intersection this probability is considered sufficient to initiate the delivery of support.

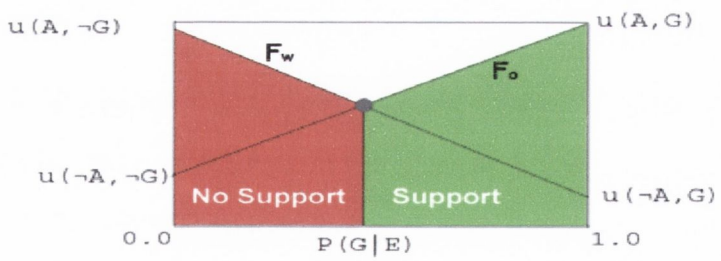


Figure 3.3 – Computing Utility Thresholds in Mixed Initiative Systems

Consider the simple graph shown in Figure 3.3. The vertical axis represents a utility value based on an observed action A and a particular goal G, while the horizontal access is a measure of the probability that a particular goal should be supported based on the evidence. Two functions are represented on the graph: F_w and F_o. F_w is a function that assesses the desirability of withholding support

while F_o represents offering support. As the probability that support should be offered increases, the desirability of withholding it drops until a point is reached after which it is preferable to offer support.

This threshold represents the point at which the marginal utility of offering support exceeds the desirability of withholding it. In mixed initiative systems, this is the point at which the system would offer support.

In the Lumière project, Bayesian Networks were used to interpret user activities, such as where the user's focus appeared to be, or their rate of typing or length of pauses, as well as monitoring and analysing the words they were typing. The project identified forty routine activities in Microsoft Excel, and by observing the user the system attempted to match their behaviour to one of these activities.

Mixed initiative systems have also been proposed as an approach to next generation Intelligent Tutor Systems [Jordan 2002], conversational recommender systems [McGinty, 2003] and interactive knowledge acquisition systems [Witbrock, 2003].

3.3. RESOURCE REPRESENTATION & REASONING

In Chapter 2's exploration of pervasive computing, the need to model and configure devices was noted. Meeting this need raises a number of issues such as ensuring that the relevant data can be reliably, intuitively and sufficiently captured and that once captured, the resulting models can be made available for whatever context and configuration demands arise.

The issue of representing knowledge in highly distributed applications, such as pervasive computing, where data needs to be meaningfully shared between entities that may not have a persistent or long-term relationship is an active area of research.

To address this, "The Semantic Web" [Berners-Lee, 2001] proposes that computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning. This paper and the W3C Semantic Web project which it initiated has focussed on developing schemas for extending knowledge through semantics, and the provision of tools that enable automated reasoning and classification of data to take place based on 'understanding' data through these semantics.

This section will firstly examine the Semantic Web's work on schemas, specifically the development of ontologies as a framework for defining custom domain-specific semantics. It will then go on to look at some of the reasoning tools that have been developed to act on these ontologies and present some applications that allow ontologies to be designed.

3.3.1. Ontologies

An ontology is a specification of a concept [Gruber, 1993]. An ontology is therefore a framework that allows users to describe an entity and its relationships and references to other entities. This referential approach means that knowledge can be shared with its meaning intact. This is useful in the pervasive computing domain, as a wide variety of different devices from different vendors have to co-operate and share information [Lewis, 2004].

Ontologies are one method for creating interoperable models. Another method would be Topic Maps (ISO/IEC 13250), but their emphasis is more on the 'findability' of data [Grønmo, 2002] rather than its dynamic classification. In addition, the Semantic Web has produced a number of powerful and well-supported tools that enable the authoring and testing of ontologies.

A commonly cited reference example of an ontology is a specification for wines. This ontology captures relationships between different types of wine on the basis of geographical origin, grape type, taste and character and so on. The wines can then be classified into new groups whose members are defined according to combinations of conditions not known when the original

specifications were defined. For example, a wine taster could then search for a particular variety according to these metrics, e.g. "Give me a Californian shiraz". Alternatively, a new classification could be defined at some time in the future such as "Wines that accompany fish". Members of this taxonomy might be dry white wines. A sommelier could consult the "Wines that accompany fish" taxonomy and see members who were defined without prior knowledge of that classification.

Ontologies have been identified as a useful tool in complex systems where knowledge needs to be captured and shared between different actors, components or independent systems [Uschold, 1999]. In such systems, different users may be responsible for a variety of tasks including ontology authoring, data authoring (the instantiation of the ontology), application developers who will make use of available ontologies and knowledge workers who make use of the information that is made available.

Ontologies have become popular in the Semantic Web community as a method for structuring data in a meaningful way, such that it can be shared among different applications within a common discourse community. The Semantic Web promotes the development of tools that allow data on the web to be extended and linked in a way that allows machines to understand the meaning rather than just the data in isolation.

To assist with this, a number of different ontology languages have been proposed, along with tools to make them useful. One of these, the Web Ontology Language (OWL), is discussed below, however there is also a review of DAML+OIL and RDF included Appendix IV.

When used with suitable Description Logic reasoners (see below), ontologies can also be used to drive the configuration of resources [McGuinness, 2003]. In these scenarios, an understood set of semantics can be used to impart the configurations to the resources, while internal semantic constraints can be used to ensure that the models are sufficiently complete such that they can be used for effective configuration.

3.3.2. Web Ontology Language – OWL

The Web Ontology Language is an improved version of DAML+OIL, sponsored by the W3C. Many of the features put forward in DAML+OIL are still present in OWL, along with more vocabulary for describing properties and classes.

OWL has three increasingly expressive sub-languages that can be chosen depending on the level of demand of the application. These are shown below in Table 3.1.

OWL Lite	OWL Lite is aimed at users who only wish to develop taxonomies. It therefore supports carrying out simple classifications based on simple constraints, e.g. in OWL Lite cardinality is limited to 0 or 1.
OWL DL	The DL in OWL DL comes from description logics. OWL DL provides the full set of OWL expressions; however it limits their possible uses. For example, while a class might inherit from many parent classes, OWL DL enforces that an instance can only be of one class.
OWL Full	OWL Full allows the author full freedom in defining their ontology. However given that this could result in objects simultaneously existing in multiple ways, there is no guarantee that a solution will present itself when reasoning about OWL Full ontologies.

Table 3.1 – The OWL Set of Languages

As the sub-languages become more expressive, they retain all of the features of their simpler forbears.

OWL's particular strength lies in its ability to enforce useful logical conditions such as cardinality, existential and universal quantifiers disjoint relationships,

symmetry, transitive relationships and multiple inheritance. With this set of conditions OWL ontologies can be checked for consistency, classified into taxonomies and the individuals they describe can be automatically classified to classes for which they were not explicitly assigned. One of the most useful ways that OWL allows such classification of individuals is by means of 'Defined Classes'.

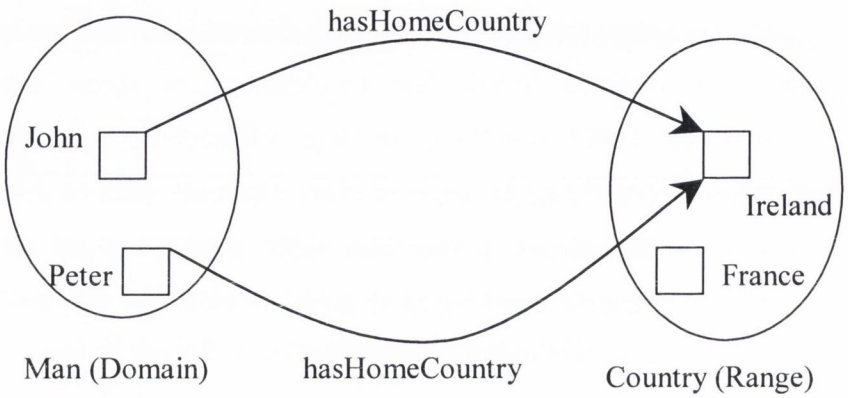


Figure 3.4 – Domains, Ranges and Relationships

A Defined Class is one whose members meet certain necessary and sufficient conditions. For example, take the classes Man and Country shown in Figure 3.4, where Man has a property hasHomeCountry whose range is Country. This relationship can be defined as functional if each individual of Man must have only one home country. This does not necessarily make the reverse proposition true that every country need only have one man.

Now consider a sub-class of Man called IrishMan. Members of this class must satisfy the condition:

```
hasHomeCountry == Ireland
```

If this condition is defined as both necessary and sufficient then any individual with the home country Ireland can be classified as a member of the IrishMan class.

In such a way, individuals of basic or primitive classes, which do not assert any necessary and sufficient conditions, can be combined and classified by a reasoner in order to make use of their data. Such reasoning is a powerful capability when sharing data, as individuals can be inferred to be members of classes for which their original author had no knowledge.

3.3.3. The SOUPA Ontology

One of the key benefits of ontologies is that by actively supporting the inter-relationship of concepts they also enable extensions of simple ontologies into more complex ones that share the same basic semantics. SOUPA, Standard Ontology for Ubiquitous and Pervasive Applications, is an OWL-defined set of ontologies aimed at supporting ubiquitous computing applications. The development of the SOUPA ontologies is being led by the Semantic Web community and in particular the Semantic Web in UbiComp Special Interest Group, SW-UbiComp-SIG.

According to its developers, "{SOUPA} is expressed using the Web Ontology Language OWL and includes modular component vocabularies to represent intelligent agents with associated beliefs, desires, and intentions, time, space, events, user profiles, actions, and policies for security and privacy" [Chen, 2004].

The SOUPA ontologies define a minimum set of inter-related ontological bases for defining the various aspects that need to be considered in UbiComp applications. These basic ontologies can then be extended for given applications, or where necessary concepts can be referenced to create new ontologies where the existing ones are not sufficient.

The ontologies themselves are based on a number of existing projects including the Friend-Of-A-Friend ontology (FOAF) [Brickley, 2003], DAMLTime and the entity sub-ontology of time [Pan, 2004], the spatial ontologies defined in OpenCyc [Lenat, 1990], Regional Connection Calculus (RCC) [Randell, 1992],

COBRA-ONT [Chen, 2003], MoGATU BDI ontology [Perich, 2004] and the Rei policy ontology [Kagal, 2005].

SOUPA Core	SOUPA Extensions
Person	Meeting
Policy	Schedule
Action	Knowledge
Time	Location
Geo-Measurement	Region Connection Calculus
Space	Device
Belief-Desire-Intention	Document
Agent	Digital-Document
Event	Image Capture

Table 3.2 – The SOUPA Ontologies

In order to make SOUPA usable, it is divided into eighteen individual ontologies and these are further divided into a core set of nine and an extended set of the remainder. These ontologies are listed in Table 3.2.

Person

The Person ontology, Figure 3.5, captures a number of aspects of a person from simple biographical details shown in Figure 3.6, to other more peripheral information such as their homepage and instant messenger provider.

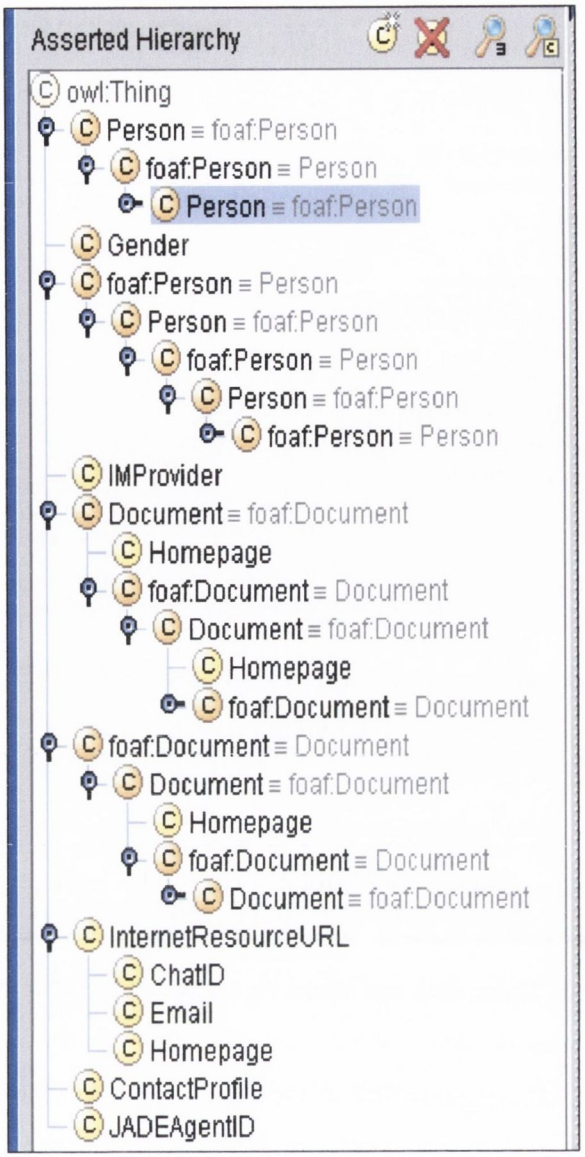


Figure 3.5 - The Classes in the Person ontology

The 'foaf:' namespace qualifiers in front of some of the classes above represent classes that reference the Friend of a Friend class. This namespace approach allows for more elegant code and is present in RDF and OWL.

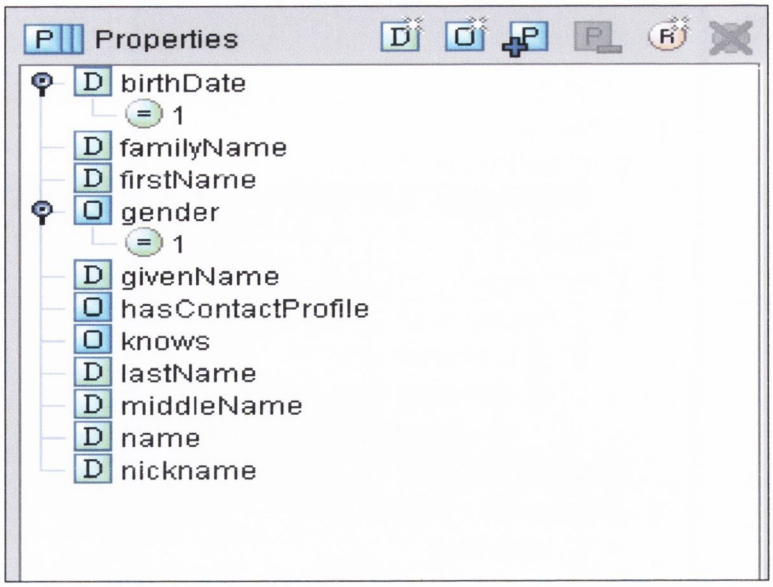


Figure 3.6 – Biographical properties of the Person class within the Person Ontology

Policy

The Policy ontology captures information such as who enforces the policy and what actions it permits or forbids. It also carries a property denoting when the policy was created. These data are based on instances of other SOUPA classes such as the creation date being an instance of the time:Instant or time:InstantEvent classes specified in the Time ontology. Such limits on the type or class of data being used can be specified using standard OWL predicates such as defining the range from which values must be chosen or the cardinality of the quantity of permitted instances of a given property, e.g. there may be only one creation date and it must come from the above classes.

Action

Actions represent high-level activities in the UbiComp environment. The properties of the Action ontology include the entity that performs the action, the entity that receives the effect after the action is performed, the object that the action applies to, the location at which the action is performed, the time at which the action is performed and the thing that the actor uses to perform the action. Action and Policy are closely related and via the permissions that Policy applies to the actions defined from the Action class.

Time

As with any distributed system, consistent timing is essential in order to keep all elements functioning in tandem. The Time ontology provides a number of time-related classes that can be used to provide measures of time. The classes include notions such as intervals and instants as well as higher-level notions such as the TemporalEvent class, which can capture more complex ideas such as endsBefore or startsAfter, thus allowing richer timing ideas to be shared.

Geo-Measurement

The Geo-Measurement ontology provides basic measurement terms such as area, direction and co-ordinates, as well as higher-level notions such as Geo-PoliticalEntities.

Space

The Space ontology is designed to support reasoning about the spatial relations between various types of geographical regions, mapping from the geo-spatial coordinates to the symbolic representation of space and *vice versa*, and the representation of geographical measurements of space.

Belief-Desire-Intention

Belief-Desire-Intention, BDI, is closely related to the Agent class outlined below. It provides constructs for describing the state of an agent in a system in terms of its beliefs, desires and intention, that is it provides a framework for describing what the agent currently knows about its world (knowledge that may not be correct), a range of end states that it wishes to bring about and a set of plans in the form of actions, preconditions and effects that it will use to achieve these end states. The BDI also captures whether desires are achievable, non-achievable or conflicting given the current available knowledge.

Agent

An agent is any computational entity or human being within the UbiComp environment that can be modelled. This model divides along two lines; a view of the state of the agent in terms of the BDI framework and instances of any actions from the Action class that it may be considering. The BDI framework simply allows for the properties 'believes', 'desires' and 'intends' to be

populated for the given agent. The Action class is extended by the BDI notion of a plan to allow for the definition of related actions calculated to bring about the intended change in world state.

Event

The Event ontology captures all activities, schedules and sensing events.

The remaining ontologies constitute the SOUPA Extensions and they satisfy two main aims. Firstly they define an extended set of vocabularies for supporting specific types of UbiComp applications, and secondly they demonstrate how to define new ontologies by extending the SOUPA Core ontologies.

3.3.4. Reasoning Tools

The various ontology languages provide a rich meta-data vocabulary; however it is still necessary to provide tools to make use of these data in semantic web applications. Typically, these tools make use of standalone description logic [Baader, 2005] reasoners such as Racer [Haarslev, 2003], FaCT++ [Wang, 2005] and PELLET [Parsia, 2004] and provide an API for classification of ontologies, the creation of taxonomies, making assertions based on available metadata and querying over objects. Typically the reasoners are based on Lisp, while the semantic tools are written in Java; however some of the semantic tools include their own simple reasoners such as Jena.

Jena[1] is a Java framework for developing semantic web applications, and it is an extension of HP Lab's Semantic Web Programme[2]. It was initially designed for use with RDF and so it is limited to supporting semantic languages built on that format such as OWL and DAML+OIL.

In Jena, ontology models are considered as an extension of the core JenaModel. When creating a new model, the author directs Jena's ModelFactory to the URI

[1] http://jena.sourceforge.net/
[2] http://www.hpl.hp.com/semweb/

where the ontology specification is stored; so for an OWL-DL ontology, the author specifies:

```
http://www.w3.org/TR/owl-features/#term_OWLDL
```

Similarly existing ontologies can be imported as long as their language specification is available.

The Jena API is capable of using all of the OWL and DAML+OIL semantics such as capturing an ontology's properties (e.g. subclass or inverses) as well as restrictions (e.g. cardinality, universal qualifiers).

Reasoning Engine	Purpose
Transitive reasoner	Provides support for storing and traversing class and property lattices. This implements just the *transitive* and *symmetric* properties of rdfs:subPropertyOf and rdfs:subClassOf.
RDFS rule reasoner	Implements a configurable subset of the RDFS entailments.
OWL, OWL Mini, OWL Micro Reasoners	A set of incomplete implementations of OWL Lite
DAML micro-reasoner	Used internally to enable the legacy DAML API to provide RDFS scale inferencing similar to that available in Jena1.
Generic rule reasoner	A rule based reasoner that supports user-defined rules. Forward chaining, tabled backward chaining and hybrid execution strategies are supported.

Table 3.3 – Jena Reasoners

Jena also allows ontology developers to make use of reasoning engines, although they do not have to be enabled if they are not needed. These engines

allow additional information to be inferred from assertions specified in the classes' metadata. The Jena package comes with a number of reasoning engines including those shown in Table 3.3.

In Jena, reasoning about ontologies is based on the Open World assumption. The open world assumption states that if a condition cannot be asserted as true from the available data, it is considered true, while a closed world assumption assumes the opposite. If a particular approach to world assumptions is required then care must be taken when choosing a suitable reasoner.

For example consider an ontology that stated that every person must have one name and one date of birth, and an instance of this ontology was created as follows:

```
Name: "Peter Smith"
DOB:  <null>
```

In this instance we have fulfilled the one name requirement, but we have not included a date of birth. If we ask a reasoner to examine this instance, using the conditions we have defined, we receive the contradictory results shown in Table 3.4, depending on the nature of the world assumption we are using.

Open World:	model is complete, there is one name and one date of birth, it just hasn't been revealed yet.
Closed World:	model is incomplete as no date of birth has been included

Table 3.4 – Open World versus Closed World Assumptions

The choice between open world and closed world assumptions is a current source of discussion in the semantic web community, and an inability to select your preferred assumption is a limitation of the Jena system.

3.3.5. *Protégé*

Protégé[3], see Figure 3.7, is an ontology editor and knowledge-base framework. It has become one of the most popular ontology editors and simple query tools.

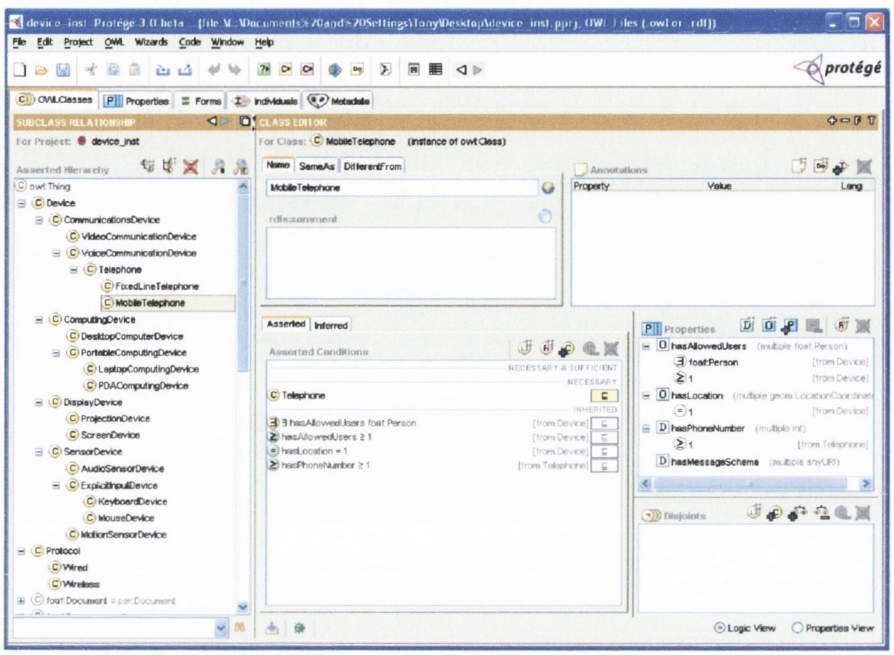


Figure 3.7 – The Protégé Ontology Editor

The Medical Informatics Group at Stanford University developed Protégé, and it is available as a free open source application, or as a set of documented Java libraries. The design objectives of Protégé are to provide an extensible knowledge model, customisable output file formats, a customisable user interface and an extensible architecture to facilitate easy integration into other applications [Noy, 2001].

This highly flexible and customisable approach has allowed developers to release a range of useful plugins for Protégé that allow it to be used for specific ontology languages or applications. Some of the most popular plugins include ones for DAML+OIL, UML and OWL.

[3] http://protege.stanford.edu/

Protégé's internal knowledge model is similar to many of the other semantic web languages. Models in Protégé consist of classes, instances of these classes, slots representing attributes of each class and facets giving additional information on the slots.

Take for example a new datatype called 'Wine' which has an attribute 'hasColour'. This attribute can have the values 'Red', 'White' or 'Rosé'. In OWL the datatype is called a class, the attribute is called a predicate and the value limitation is known as a constraint. In Protégé these are known respectively as the class, slot and facet.

The graphical tool provides a very efficient way of designing ontologies and manually creating instances. In addition, Protégé's source code supports the development of custom applications that extend on its core functionality.

3.3.6. Protégé-OWL

As discussed above, Protégé is equipped with a number of powerful plugins that allow the platform to be tailored including Protégé-OWL plugin [Knublauch, 2004]. From a user perspective the most immediate difference is that the interface changes to reflect OWL specific requirements and naming conventions.

As with the main Protégé application, the Protégé-OWL plugin is available as an open source Java package, and again it provides a number of useful features for application designers. In particular, the Protégé-OWL plugin offers developers a set of in-built reasoning tools that offer a closed world approach to reasoning.

The Protégé-OWL plugin has many similarities with Jena, and many of the concepts in its knowledge model can be easily mapped to similar concepts in Jena. For example, the Jena ontology class *OntModel* is replaced by the *OWLModel* class in Protégé-OWL; however the key difference is that Jena

adopts an open world assumption, while Protégé-OWL operates on a closed world basis.

3.4. AUTONOMIC COMPUTING & GOVERNANCE OF PERVASIVE COMPUTING

With computing systems becoming ever more complex, and reliant on a proliferation of interconnected devices, it is becoming necessary to develop new management paradigms for systems. With potentially millions of devices interoperating, any solution requiring the individual configuration and management of each device is not feasible [Kephart, 2003]. This is a particular issue in governing pervasive computing systems where there is a proliferation of devices.

An approach to this management problem being promoted by IBM is Autonomic Computing [Kephart, 2003]. As a leading supplier of complex computing systems, IBM have identified that the capital cost of large systems is often surpassed by the ongoing cost of administering and maintaining the systems. Their motivation for proposing an autonomic solution is that it allows for the systems to self-configure, self-optimise, self-heal and self-protect thereby reducing the ongoing cost of maintenance and enabling the integration of new components with minimal human involvement.

Autonomic Computing takes its lead from nature. The autonomic nervous system allows cells to perform routine tasks such as digestion or respiration without constant intervention or instruction from the parent organism. This allows cells or groups of cells to carry out these routine functions by consulting their genetic code and responding to changes in their environment.

Researchers have proposed that the autonomic approach could be replicated in computing systems whereby policy goals are set for the system, but implementation is delegated to the individual components of the system.

Such policies could include 'Maximise Efficiency.' The multitude of devices could receive this policy statement and attempt to satisfy it by whatever means they individually deem appropriate. Such policies could therefore be prescribed for a suite of devices, which would then attempt to satisfy the aims of the policy without any further user intervention. This has similarities with Policy Based Management, however it differs insofar as all responsibility is delegated to the components and the policy author need not have any knowledge of what resources will be accessing the policy.

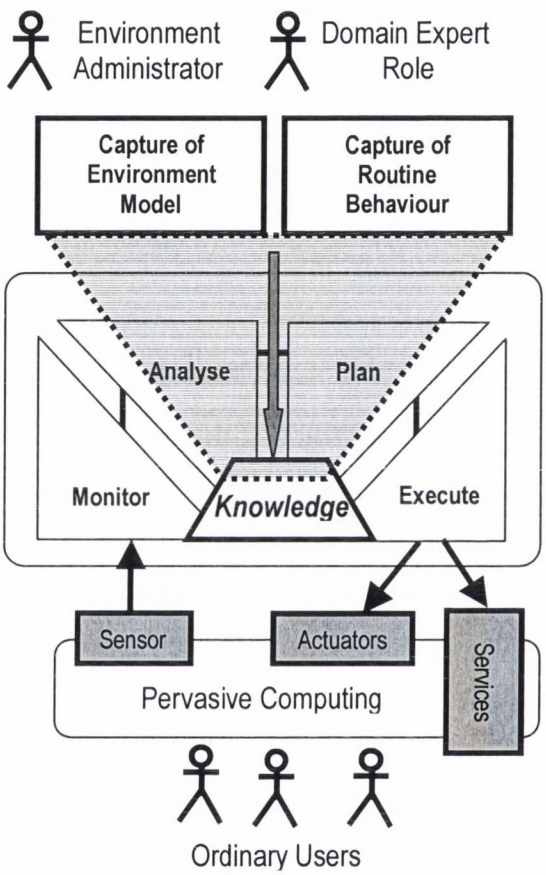


Figure 3.8 – A view of the proposed architecture using the IBM Autonomic Computing model

From a pervasive computing perspective, autonomic computing provides a useful approach to managing massively prolific devices. If one were to imagine the potential application of autonomic systems as a continuum with wholly automated systems at one extreme and highly user dependant systems at the other, the governance approach proposed in here would fit somewhere in the middle. It does not take all control away from the user, however using a mixed

initiative approach it attempts to minimise the need to interact explicitly with the user.

Figure 3.8 demonstrates how the governance approach discussed in Chapter 2 can be applied to the IBM conventional representation for autonomic computing.

The approach differs from much of the current research in autonomic computing which tends to focus on system requirements rather than human user ones, [Iqbal, 2005] [Gelenbe 2005] [Boutilier, 2003] [Pacifici, 2003]. The approach here instead proposes to leverage this research work and by combining it with a user-centric interface technology, provide a platform from which services can be offered in pervasive computing environments.

3.5. SUMMARY

Chapter 2 established the functional demands that the proposed approach to governance would make on a prototype. This State of the Art review has examined some of the technologies and tools that can be used to develop prototypes that seek to meet these demands, and also provide a means for evaluating the proposed approach to user centric governance. These technologies are summarized in Table 3.5 below.

Requirement	*Technology*
Inference	Mixed initiative using Bayesian Networks (JavaBayes)
Environment Modelling	OWL-based models extending SOUPA and reasoning with Protégé-OWL
Routine Behaviour Descriptions	Utility-based Bayesian Networks

Table 3.5 – Summary of Governance Requirements and Proposed Approaches

4. DESIGN & IMPLEMENTATION

4.1. INTRODUCTION

4.1.1. Overview

Chapter 2 presents a discussion of the demands posed by user-centric governance of pervasive computing environments, and Chapter 3 offers a review of some technologies that could assist with meeting these challenges. Specifically an integrated approach to supporting users through inference and a governance approach to managing this inference have been proposed.

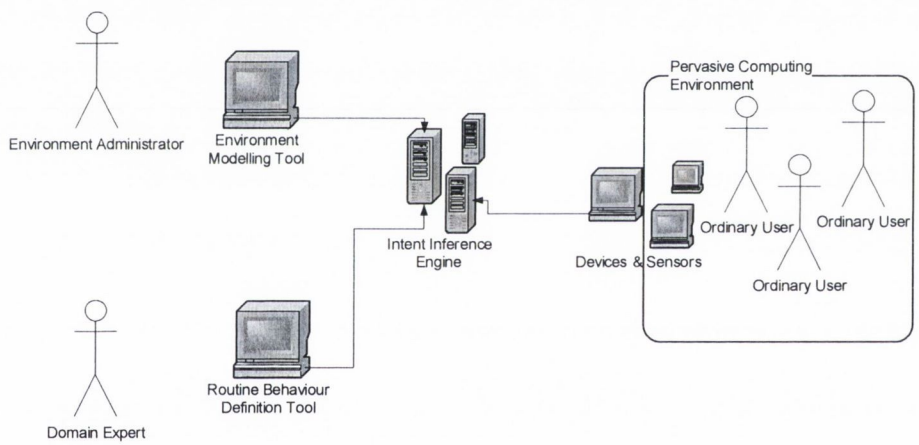


Figure 4.1 – User Centric Governance Architecture

Such an integrated system is presented in Figure 4.1 and reflects the three-way design approach that will be covered in this chapter. This approach identifies three user roles that are involved in the target pervasive computing environment, namely: ordinary users, domain experts and environment administrators.

The ordinary users are the routine users of the environment who expect that by interacting and manipulating the environment they will receive appropriate

system support. Their interaction with the inference engine is handled via sensor data and relevant context.

The domain experts have knowledge on routine behaviour in their target environment and can use this knowledge to help govern the inference process. The Routine Behaviour Description Tool (RBDT) provides a means for this knowledge to be imparted.

The environment administrators are responsible for developing models of the devices in their environment that can be used to provide context data for the inference process and configuration requirements for the modelled devices. They do this via the environment modelling tool.

This chapter will present a process analysis for each component that will identify its inputs, outputs, assumptions, objectives and actions. These will then be examined to see how a potential design might meet their demands. This examination will result in some experimental objectives that an implementation will have to address. The proposed prototype implementations will then be presented.

The first component examined is the Intent Inference Engine, as this was the first to be developed. It is also the point of integration for the other components and it therefore sets the requirements for the others. In practice however, the Environment Modelling Tool and the Routine Behaviour Description Tool would be used in advance to configure the inference engine.

4.1.2. Chapter Structure

This chapter takes the design requirements outlined in Table 3.5, and examines how to create components to meet each challenge. This structured examination begins with a Process Analysis of each system based on high-level representation of the architecture of each component. This analysis looks at the Inputs, Outputs, Assumptions, Objectives and Activities for each component. These are then used to derive specific prototype objectives.

The second half of the chapter takes these prototypes and describes an implementation for each. Each description finishes by returning to the architecture diagrams from the design and reproduces them showing the various technologies used in the implementation.

4.1.3. Overall Design Assumptions

A number of design assumptions have been made when considering how to address the needs of the three roles identified above. It is assumed that the target environment is equipped with a suitably rich set of input devices and sensors that can reliably report on user activity. These devices are assumed to monitor a user's position, their manipulation of objects within the space, the user's identity, their speech and prosody and any other relevant data that might be of interest in inferring user intent.

It is also assumed that where specific expertise is required to produce input artefacts (ontology definitions, maps of the target environment); this will be done outside the scope of the prototype components. The actors responsible for producing these input artefacts will be clearly identified in the designs.

4.2. INTENT INFERENCE ENGINE

Chapter 2 discusses the need to provide a mechanism for aggregating input data and making inferences about a user's intentions. It also notes that to be truly effective, such a process would need to be governed in order to ensure that the support it recommends is appropriate.

4.2.1. Process Analysis

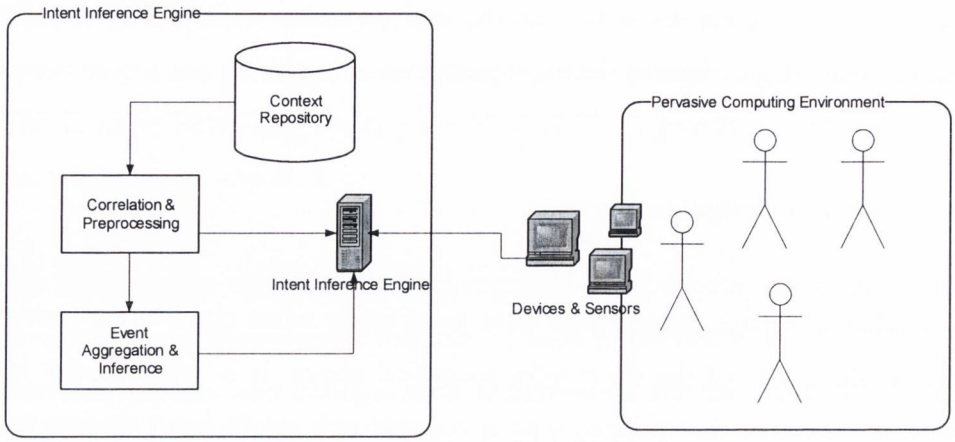


Figure 4.2 – Intent Inference Engine

Based on the design in Figure 4.1 and the functional requirements outlined for the ordinary users in Chapter 2, Figure 4.2 provides a view of the components that make up the intent inference system.

Based on this design, the following concerns can be identified:

Inputs

As discussed in Chapter 1, the inference process has two types of input data: usage information and governance information. The usage information is based on user activity and context. This data needs to be incorporated into the inference system's view of the world so that decisions regarding pre-emptive support can be made.

Governance information concerns the data that is used to manage the inference process. In the case of the design proposed here, two types of governance data have been identified namely behaviour models and environment models.

Outputs

The output of the inference process is a belief regarding the type of support that the user requires. The output is not the support itself, but rather information that can then be used by a suitable service composition engine to deliver this support on demand.

Assumptions

In general, it is assumed that target environments will be equipped with sufficient sensors and context repositories to deliver necessary input data. It is also assumed that a pre-processor can be offered that parses the input data so that it can inform the inference process.

However, for the purpose of the experimental examination presented here, no real sensors will be used, so the main experimental assumption is that by defining suitable pre-processor outputs, the prototype does not need to be concerned with how the input data is sourced (simulated data versus sensed data, etc.).

It is also assumed that the system will only operate on a single user basis. This assumption will allow the core inference mechanism to be evaluated without introducing additional concerns regarding collaboration.

Objectives

The objective of this component is to provide a mechanism for aggregating input data, and delivering utility-based inference on user intentions for a single user in a simulated environment.

Activities

The system needs to develop an inference model of activity in the space. This model reflects sensed activity and context in the space, and should provide a means for relating data so that higher-level intentions can be inferred. In order to achieve this some activity concerns need to be considered such as:

- Pre-processing input data
- Identifying an inference mechanism
- Self-learning – can the system facilitate self-improvement over time

4.2.2. Prototype Objectives

In a pervasive computing environment there are a number of forms of input data. This data can be broadly divided into two types, namely sensed data and context data. Some examples of each are shown in Table 4.1.

Sensed data comes from sensors in the space and can come in a near continuous stream as users move around and interact with each other and the space.

The sampling interval for context data may be less frequent. Much of this context data remains relatively static for the duration of a particular activity, for example the device model is unlikely to change over the course of an event, whereas data on roles may be more fluid as users often change roles over the course of a particular event.

Input Data Type	Examples
Sensed Data	Motion sensor data Audio sensor data RFID tracking
Context Data	Time Date User profile Device Profiles Temperature Access privileges Roles

Table 4.1 – Types of Input Data

In order to make use of input data, it first needs to be pre-processed so that the inference process can respond appropriately by firing the required evidence nodes. This is of particular importance when dealing with heterogeneous input streams. The nature of this processing depends on the complexity of the input data. If the data is continuous it may be necessary to timestamp it so that it's usefulness can be viewed as a function of its 'freshness' – data regarding an action an hour ago may not be as informative now as it was at the time.

Consideration should also be given to the ambiguity of the input data, especially when recording activity. Certain types of sensor may be less accurate than others, and this should be reflected in how confidently their data is handled. There are two approaches to this problem. One is to weight the data as it comes in so that the less confidence there is in the data, the less information it is judged to contain and consequently the less effect it has on the inference process. A second approach is to build this uncertainty directly into the inference model by setting its own internal weightings to reflect the reliability of the data at the node recording that input stream.

The approach proposed here follows the second option. These sensor weightings are built into definitions of high-level concepts, which map combinations of sensor data to higher-level human activities like "standing by the window" or "sitting in the audience".

Conversely, context data should be richer and less ambiguous as calendars and profiles provide more rigorous and scrutable data models. As a result, context data with a high confidence rating can be considered definitive. For example, an inference model may require that certain resources are available in order to allow a given outcome, e.g. a display device must be present so that a map can be displayed. Unambiguous context information such as 'no displays are present' can immediately discount the possibility that a map can be displayed thus saving on future inference.

Implementing a comprehensive solution for context would be a significant research challenge on its own. In the design proposed here, context information such as the nature of available devices can be used to immediately discount

certain types of support. However more complex context support is not addressed.

The requirement here is therefore to provide a suitable mechanism for pre-processing available input streams to make this data usable as evidence for the inference process.

However, once that data has been suitably processed, it still has to be aggregated and responses made. The Microsoft Lumière project [Horvitz, 1998] identified a similar need to perform utility-based analysis on processed data. Researchers wanted to infer users' intentions as they made use of the Microsoft Excel application. The researchers identified 40 routine tasks that a user might perform. They also selected a number of ways of capturing user behaviour including tracking the words users were typing and timing pauses in activity. By developing a model for correlating combinations of input data to the 40 tasks, they were able to reliably offer support to the user on the basis of their recorded behaviour. The aggregation and inference technique used by the Lumière researchers was based on Bayesian Networks.

As described in Chapter 3, Bayesian Networks are a proven technique for aggregating observed data to reach high-level conclusions in a range of applications from calculating infant mortality rates in Burkina Faso [Sankoh, 2002] through to selecting the optimal stock portfolio [Rossi, 2002].

The requirement here is therefore to design some tool for interpreting Bayesian Networks, adding knowledge to them and carrying out inference in order to derive high-level outcomes.

The final action noted in the process review is self-learning. In the examination of governance challenges presented in Chapter 2, a need for self-learning was identified, especially in less formal environments such as in the home. However, this need has been placed beyond the scope of what is examined in the prototype proposed here.

4.3. ENVIRONMENT MODELLING

Chapter 2 identified the need to develop an environment-wide model of the devices that could provide context input to the inference process. It was also noted that if these models were appropriately defined they could also be used to configure the modelled devices. The recommended approach was therefore to develop semantically complete configuration models that could also be used as a context input for inference.

4.3.1. Process Analysis

In the overall system view in Figure 4.1, the environment modelling tool is responsible for capturing device models. Figure 4.3 shows the system elements that make up the modelling tool.

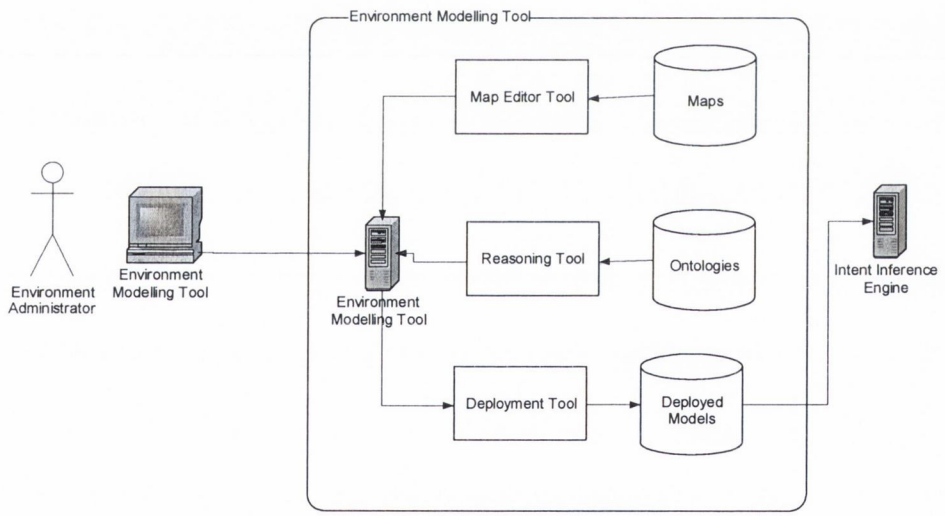


Figure 4.3 – Environment Modelling Tool

Inputs

The inputs to the modelling tool can be divided into two types. The Map Designer and the Devices Designer are system experts, while the Environment Administrator is part of the routine governance of the system.

The Map Designer is responsible for preparing the 3D view of the target space. These maps can then be used by the Environment Administrator to visualise the space they are working with.

The Device Designer prepares the ontology definitions, including necessary constraints. These are instantiated by the Environment Administrator by using the modelling tool.

Outputs

The output is a semantically complete model that can be used to provide context data for inference and can also be used to configure devices.

Assumptions

It is assumed that there are suitably qualified map and device designers who can prepare the necessary input models.

It is also assumed that the devices in the space are capable of interpreting and responding to the configuration models.

Objectives

The objective of this component is to enable users to intuitively create semantically complete models of the devices in the target environment such that these models can be used to provide context input data for inference and configuration requirements for the modelled devices.

Actions

The actions in this design are as follows:

- Users work with the 3D map to define the devices that they want to equip their target space with

- The output of this map editor is converted to a model that can be used for semantic reasoning
- The system must direct users to add whatever additional data is required by the system to create semantically complete models
- The produced models need to be deployed

4.3.2. Prototype Objectives

As has been discussed, the user should not have to acquaint themselves with the configuration requirements of every device in the space. They should instead be able to define regions within which the target devices must adapt to meet a particular set of requirements. For example, the user might define a zone around a lectern in an auditorium and state that laptop computers within that space have control of projection equipment. A laptop entering this space would then be able to take control of the projector without any direct configuration by its owner.

The devices in a ubiquitous computing environment could range from the very simple such as cellular telephones through to multimedia systems like desktop computers. As the capabilities of the device on which the configuration system runs will limit its functionality, it is assumed that the configuration programme will be hosted on a media rich terminal device and so will not be limited by restricted resources. This allows for a graphical interface to be used, which is important, as graphical tools have been shown to be more intuitive than text-based systems, and in particular users find appropriate 3D representations of knowledge easy to interact with [Chalmers, 2001].

The design requirement is therefore to provide users with a graphical tool, which uses a 3D representation of the environment to enable users to quickly and intuitively make decisions about resources in the space.

Human beings find it easy to express their preferences and requirements when they are equipped with the language and tools to do so. For example, people can express complex desires such as the particular make, colour and trim of a

car without having to become engaged in the business of delivering the components and labour that will make this desire a reality. This abstraction focuses the user on their high-level requirements without requiring them to involve themselves in the underlying processes that make their desire a reality.

Similarly with computer systems, a novice user might have an idea of the tasks they want the system to perform, but they would not express these requirements in the language of the veteran programmer. This is an important concern when designing a configuration tool such as the one proposed here. It should, as much as possible, convert device requirements into easily understandable concepts that do not require any expert knowledge of the target device. If the configuration tool is appropriately designed, most routine functionality should be exposed in the tool.

The requirement here is therefore that system level device requirements should be translated into more accessible concepts that a novice user can understand, and that the system should be able to relate these requirements to the user in the simplest and most accessible fashion possible. This approach will allow for the system to use the terms as part of a meaningful dialogue with the administrator when trying to develop sufficient configurations.

In a pervasive computing environment it is expected that there will be a proliferation of heterogeneous devices, and that these devices will likely come from many different vendors. This means that each device may be running its own internal operating system and have its own internal configuration requirements. This would complicate the indirect configuration of the devices via environment level policies. It is therefore assumed that rather than providing a common procedural interface, the devices provide a declarative interface that describes their requirements using a common vocabulary and that in turn they can accept policies written in an ontological format. This would facilitate heterogeneous devices to understand policies by having an understanding of their shared semantics. By adopting a semantically driven model, that follows a standard ontological format, the value of such models is greatly increased as they can be reused for configuration [McGuinness, 2003].

The requirement is therefore to select an appropriate, exchangeable ontology format, and to develop a suitable schema to allow models to be created that could then be used for configuration. There are specific ontological policy languages such as Rein [Kagal, 2005], however as the models here are concerned with simple constraints checking on shared concepts a standard ontology language is considered sufficient.

As mentioned, many current knowledge representation formats are based on ontologies and include logic for enforcing conditions on the data being used. These conditions are outlined in Chapter 3, and include cardinality, inheritance and defining existential and universal quantifiers.

The models generated by this tool will need to contain sufficient information for the local devices to adapt to the user's requirements. Potentially, if the target devices were sufficiently powerful they could attempt to do some arbitration on incomplete models, however, since not all devices will have this option, the modelling tool will instead need to ensure that deployed models are sufficiently complete for context purposes and for self-configuration by the modelled devices.

In addition, an ontology based approach will allow the system to offer feedback to the user as they populate the ontology as to whether or not they have given sufficient information for the policy to be deployed. If the device attributes are properly defined, the terms used could provide a basis for this kind of system-user dialogue.

The requirement is therefore that the configuration tool should allow users to create semantically complete models in an intuitive way. As these configurations will contain location and functional data, the tool should support the ability to meet both types of data. This will require two types of interface namely one which allows location data to be easily captured, and then a further tool that can make use of the semantics of the modelled resources to prompt the user to provide whatever additional data is required. In determining whether the models are sufficiently complete for deployment, and the system should

interact in an accessible way with the user in order to solicit whatever additional information is lacking.

These two tools should interact in as seamless a fashion as possible.

4.4. ROUTINE BEHAVIOUR DESCRIPTION TOOL [RBDT]

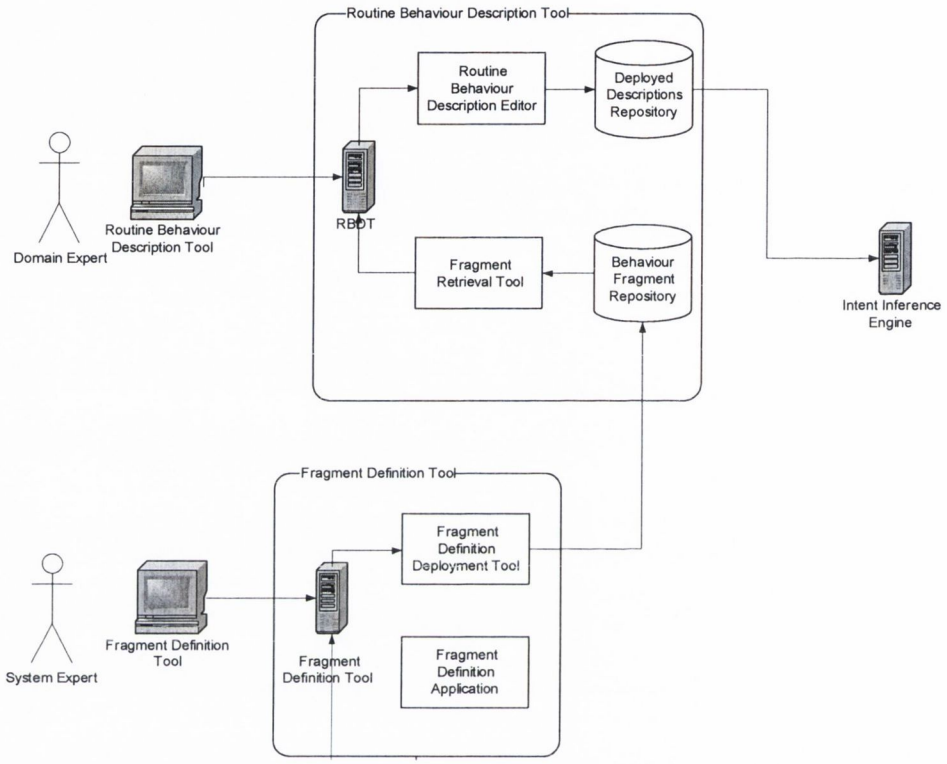


Figure 4.4 – Routine Behaviour Description Tool

The role of the domain expert requires a tool that will enable them to describe the routine behaviour in their space. These routine behaviour descriptions can then be used to guide inference and make decisions about the kind of support that needs to be offered to the users of the space. In the target pervasive environment it is assumed that there are sensors and stores of context data. The challenge is therefore to provide a means of interpreting this data.

The Routine Behaviour Description Tool should allow domain experts to associate different activities so that requests can be made for appropriate support.

4.4.1. Process Analysis

The process shown in Figure 4.4 has the following concerns.

Inputs

The RBDT has two inputs: patterns of basic activity developed by system experts and the knowledge being imparted by the domain expert via a suitable tool. In the diagram the mechanism used by system experts is labelled a Fragment Definition Tool. The fragments defined by the system expert bind input data to simple human level concerns such as 'low ambient volume', or 'standing up'. The domain expert can then associate these basic concerns to describe how these combinations relate to a user's intentions.

Outputs

The output of the RBDT is a utility-based network of nodes representing activities and edges representing their relationships.

Assumptions

This design assumes that the system experts can create stored fragments of activity that translate combinations of low-level input data into human level concepts. For the purposes of the prototype discussed here, this is a reasonable assumption as the input data is simulated and the focus of the prototype is on the method for combining concepts rather than the ways that such concepts can be defined.

Objectives

The objective of the RBDT is to produce utility-based networks that relate combinations of events to potential outcomes. These networks should be in a format that can be reasoned on by the inference engine.

Activities

The activities that this component performs are to:

- Allow users to combine events such that the links between events encode a measure of utility
- Import previously defined patterns into the current model
- Convert user designs into a format that can be used by the inference engine
- Deploy output patterns for use by the inference engine

4.4.2. Prototype Objectives

Even the simplest human activities are composed of a series of individual discrete actions, which taken in isolation do not necessarily lead to the overall outcome, they may instead indicate any number of potential activities when taken in combination with other actions. Therefore, any description of user activities should be an aggregation of simpler activities, mapping at the simplest level to sensor inputs or context data. The resulting framework therefore provides a means of taking heterogeneous input data and combining them such that the user's high-level activity can be inferred.

Such models would show how inputs could be related to high-level outcomes; however they also need to address the nature of individual relationships between concepts. For example, consider the high-level outcome C, which is based on the inputs from A and B. If A is of very high influence to the outcome, while B is less so, it would be important to capture this knowledge in the model. Such a relationship can be used to capture the notion of utility in the discussion of mixed initiative in Chapter 3.

In the model produced by the Routine Behaviour Description Tool, RBDT, it is therefore important that utility can be captured, as this will be a key factor in carrying out future inference.

The RBDT should provide an interface that can allow users to easily and quickly impart their knowledge of user behaviour into the system so that support can be offered.

Once a user has designed their configuration, the next step is to deploy this configuration. This poses two challenges: firstly some syntax needs to be identified for encoding the data, and secondly the configuration needs to be stored somewhere.

The coding syntax should be in a format that can be used by other components in the overall system, and in particular it should be intelligible to whatever system will later handle inference. Given that the eventual system will be highly distributed, this storage location will need to be available over some network, it will also need to be highly available, reliable and editable.

As in any well-engineered system, re-use of proven components is an efficient design decision. In the RBDT proven models and fragments of models should be made available so that in the future they can be re-used. For example, if Anne is preparing the weekly project meeting, she should be able to access the previous week's model, which can then be amended to meet this week's requirements.

4.5. SUMMARY OF DESIGN REQUIREMENTS

Table 4.2 presents the design requirements as discussed above for each of the components.

Component	*Requirement*
Inference Engine	O1.1 Correlate sensor and context data O1.2 Perform utility-based inference on parsed data
Environment Modelling Tool	O2.1 System should produce a single environment model O2.2 Users of varying skill should be supported O2.3 Configurations should be in an interchangeable\exchangeable format O2.4 Constraints should be used to ensure models are sufficient and complete
RBDT	O3.1 Provide a mechanism for describing routine behaviour O3.2 Facilitate the production of a directed acyclic graph linking activities (nodes) and their causal inter-relationships (edges) O3.3 Provide a requirements capture tool that minimises the cognitive load on the expert O3.4 Identification of a suitable representation format O3.5 Provision for the re-use of pre-existing activity fragments

Table 4.2 – Component-by-Component Design Requirements

4.6. IMPLEMENTATION

So far this chapter has examined the design challenges associated with the proposed governance approach. This examination has included the general issues that developing such a system poses, as well as identifying some specific requirements associated with the prototype that will be experimentally evaluated. This section looks at how to implement the prototype components, such that they meet the design requirements listed in Table 4.2.

Each implementation required considerable quantities of original Java code to be developed, and some rely on outside libraries to provide specific functionality e.g. using protégé-owl to provide reasoning capabilities.

4.6.1. Common Technologies

All of the programming code in this implementation and in the evaluation prototypes is written in Java 1.4.2, using the NetBeans 3.5 and 4.1 IDEs. Java was selected as many open source packages are available that complement the work being undertaken.

The eXist XML database [Meier, 2003] is used to store the various XML input and output data. eXist supports natively storing XML. Stored documents can then be queried using XPath. It has some subtle differences from more traditional relational databases; a particular example being how it stores entries at a given location rather than using the more conventional approach of rows addressed via keys.

4.7. INTENT INFERENCE ENGINE

This component must address the following requirements:

O1.1 Correlate sensor and context data

O1.2 Perform utility-based inference on parsed data

As mentioned in the design section, the prototype is limited insofar as it will seek to prove the approach but it will not fully explore the range of potential influences on the successful delivery of support to users. A more complete evaluation would require a significant amount of expertise in behavioural science, psychology, operations research and management science.

What is proposed instead is an open approach, which is agnostic about the origins of the data being considered. Using this approach means that future users could add their own expert knowledge to the user activity model and thus better tailor the support offered.

It is also worthwhile to mention that much of the work outlined below is built on the open source JavaBayes [Cozman, 2000] programme developed at Carnegie Mellon University.

4.7.1. Prototype Outline

The proposed prototype was specifically designed for a particular experiment, however the approach it seeks to verify can be generalised for more complex experiments.

Rather than a costly, real world experiment complete with many devices, this experiment was limited to a 2-dimensional simulator. In this simulator, mouse position was used to simulate gaze, mouse dragging represents gestures and voice input was replaced by typing in statements. Context data was made available through a device ontology that matched the positions of the devices to their co-ordinates in the image, and included functional information on the capabilities of the devices.

The 2-dimensional simulator shown to the user is shown in Figure 4.5.



Figure 4.5 – The 2-d Prototype

O1.1 Correlate sensor and context data

Input data in this experiment came from two main sources. Firstly an ontology specifying the locations and attributes of devices in the space was specified. Secondly the simulated sensor data from the typed commands and simulated gaze were used to record user behaviour.

As with the Lumière approach, vocabulary of commonly used words was prepared for use by the test subjects. This vocabulary is shown in Table 4.3.

Type	Values
Colleague's Names	Peter, Mary, Anne, Sean, Susan, Michael, Karen, Fred, Laura
Verbs	Send, Phone, Copy, Show, Call, Transfer, Display, Talk, Play, Ring, Move

Table 4.3 – Experimental Vocabulary

When phrases arrived into the system such as "I want to call John", the system takes note of the current fix of the user's gaze and correlates the input data.

Using the focus of the user's gaze, as simulated by the mouse pointer, the correlator first retrieves the ontology entry for the device. It then starts analysing the phrase entered by the user. This phrase is divided into nouns and verbs according to the dictionary of terms available to the parser. Once this is done the correlator forwards the processed input data to the inference mechanism for consideration.

O1.2 Perform Utility-based Inference on Parsed Data

As mentioned previously, the JavaBayes application was used to carry out inference. It is an open source Java application that consists of a GUI application that allows users to design and query Bayesian Networks.

For this application the inference functionality is of particular interest and so a proxy application was written which could make use of the inference libraries that ship with JavaBayes. This proxy exposes methods for loading existing models, adding information to the resulting network and querying nodes.

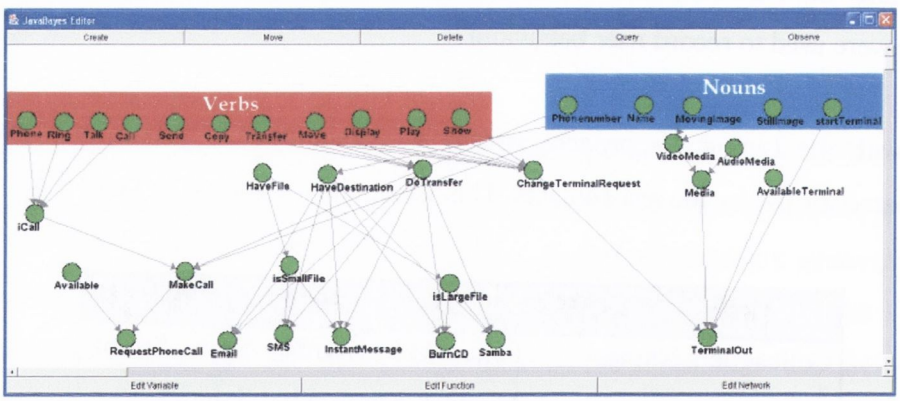


Figure 4.6 – The Bayesian Network Used

Before conducting the experiment the network shown in Figure 4.6 was defined using JavaBayes. As can be seen, the verbs and nouns used by the correlator are highlighted. As correlated information arrives into the inference engine, it is analysed and added to the current network as evidence. As this

experiment will only feature one user at a time, there is an analysis of the network with every update. In a more complex scenario, where there is a more continuous stream of input and more concurrent users, an asynchronous analysis may be more prudent given the potential computational cost of running analyses.

4.7.2. Summary

Figure 4.7 reviews the design for the Intent Inference Engine and now includes the technologies that have been identified to implement the various components.

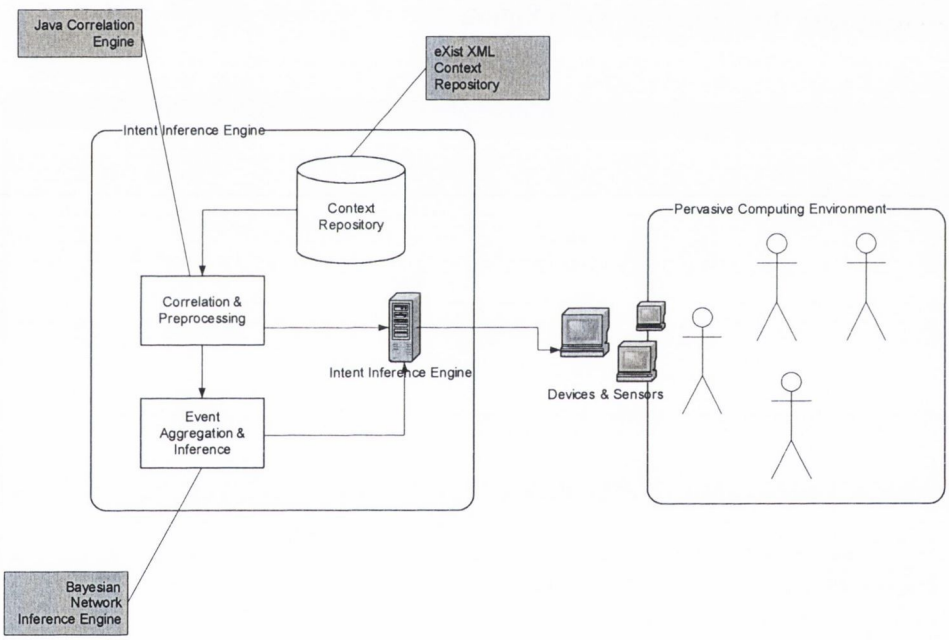


Figure 4.7 – Intent Inference Engine with Implementation Technologies

4.8. ENVIRONMENT MODELLING TOOL

In Chapter 2, the role of the environment administrator was described. This individual is responsible for modelling the resources in the space so that they can provide both configuration and context data. A number of particular design

objectives were outlined for the tool that will support these administrators. These objectives are listed below.

O2.1 System should produce a single environment model

O2.2 Users of varying skill should be supported

O2.3 Configurations should be in an interchangeable\exchangeable format

O2.4 Constraints should be used to ensure models are sufficient and complete

As noted in 4.3.2, addressing these requirements will involve the use of more than one interface. The implementation presented here uses a significant custom application to handle the majority of the solution; however a third party tool is used to assist with capturing location information. Initially users are presented with the screen shown in Figure 4.8.

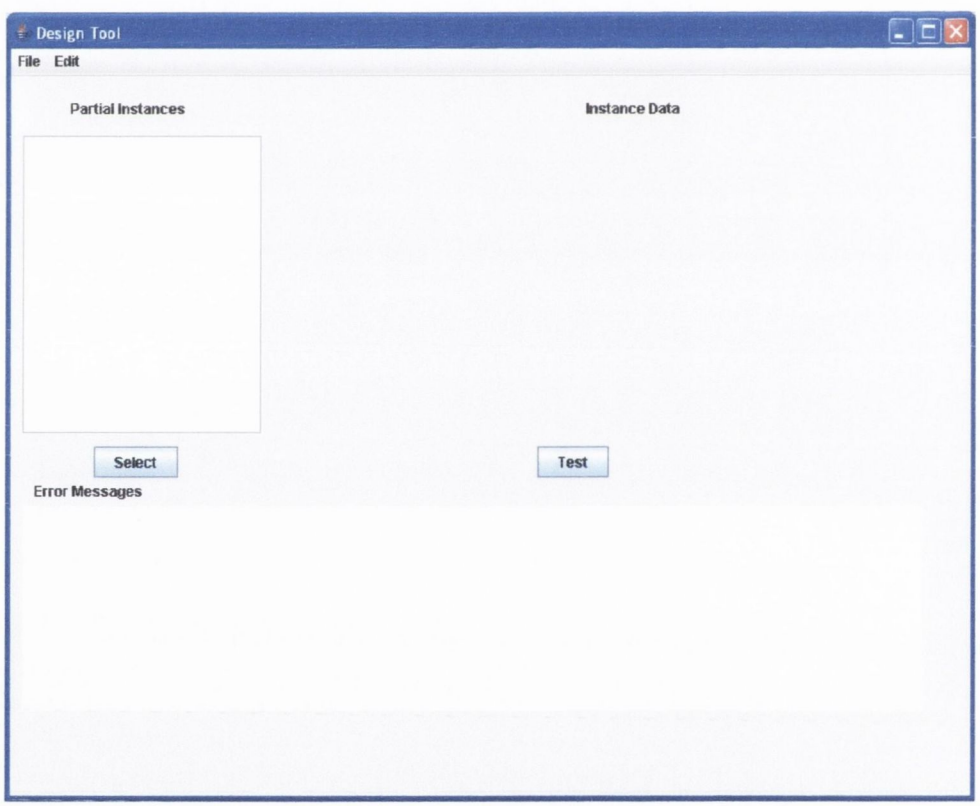


Figure 4.8 – The Environment Modelling Tool

From this screen users can either load an existing OWL model, or they can launch the 3D editor shown in Figure 4.9. On completing their graphical model, users then return to the modelling tool to complete whatever outstanding data is required.

O2.1 System should produce a single environment model

In order to meet this challenge, the environment administrator will be presented with a 3-dimensional model of the target environment. They can then interact with this model to define configuration policies for devices or areas within the environment. The system uses this to produce a single output file that can then be interpreted by the devices.

O2.2 Users of varying skill should be supported

As outlined previously, graphical tools offer an intuitive means of working with data. In order to facilitate this, a 3-dimensional editor was selected to represent the target space. A number of 3D graphical design tools are available on the market today. They are mainly designed with particular tasks in mind such as Computer Aided Design (Autocad), animation (3D Studio Max) and computer games (Hammer). Many of these programmes contain advanced functionality for their particular domain such as controlling lighting and shadows; however for the purposes of this tool the editor doesn't need such powerful features. In short, a suitable editor should provide the following:

- Ease of use
- Allow for the rapid addition of objects into a space
- Provide some methods for describing the objects with at least their type and an identifier
- Produce a regular, reusable output file
- Have minimum cost or be freely available

Version 3.4 of Valve's Hammer map editor, shown in Figure 4.9, satisfies these requirements. It has been developed for designing maps for the Half-Life computer game and allows for the rapid creation of 3-dimensional maps for use

in the game. As certain objects in the game fulfil particular roles, the editor allows for new additions to be assigned a type and any number of other attributes that are specified in an external configuration file.

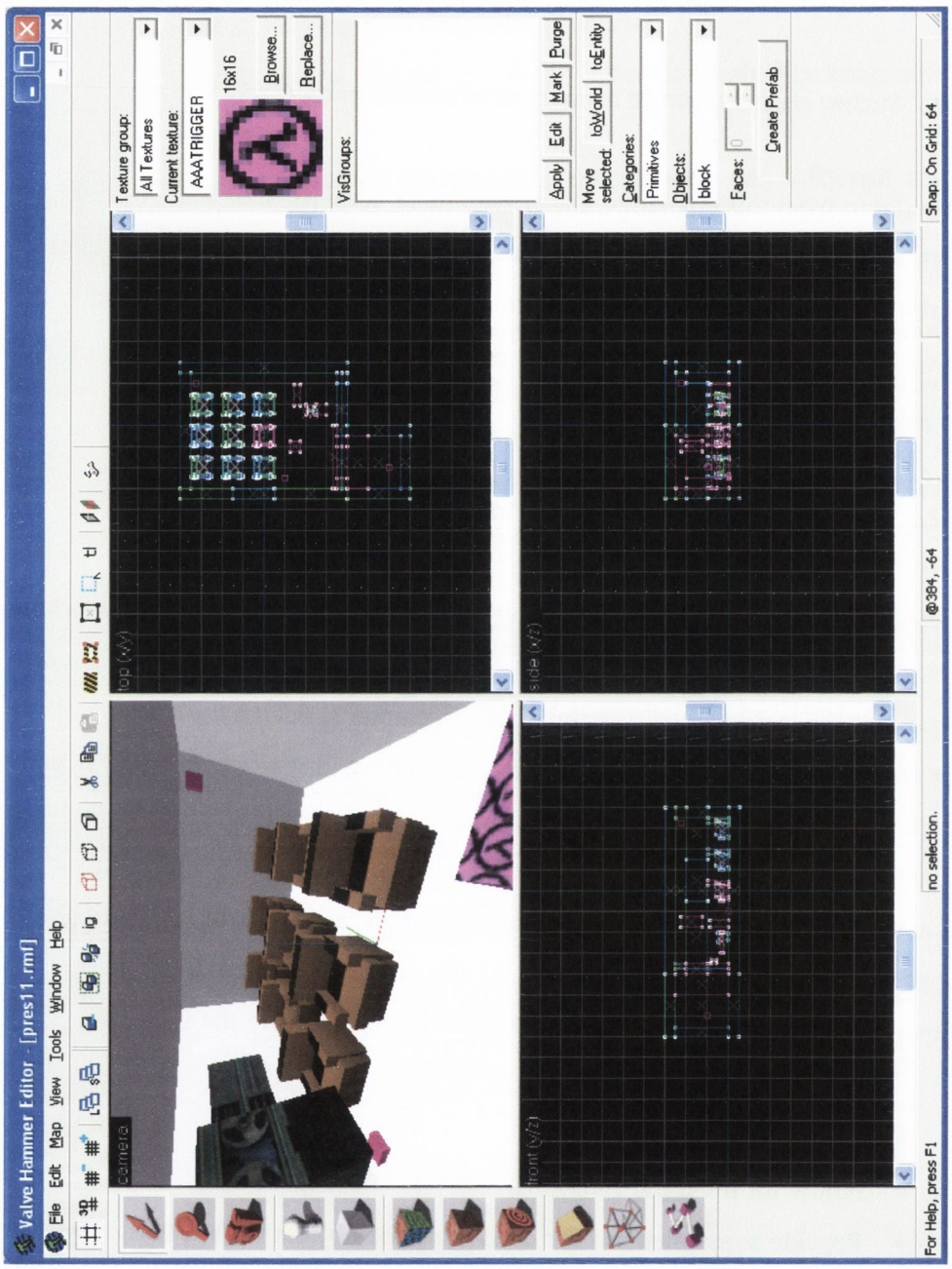


Figure 4.9 – the Hammer Map Editor

When a map has been designed it is compiled so that it can be used by the game. This compilation produces a number of files; some compile into unintelligible binary formats, while some are in regular text. One of these text files, the *.MAP format, contains information on the objects in the map as well

as their respective attributes. A snippet of a MAP format file is shown in Table 4.4. The attributes of interest here are *classname* as it defines the type of object and *targetname*, which is the name of the object. The other attributes refer to properties needed by the game; however custom attributes may also be defined.

This regular structure allows the file to be parsed and used by the other components in the system. This parsed file is in XML and is stored in a log, but is not directly displayed to the user.

```
{
"classname" "func_door"
"rendercolor" "0 0 0"
"angles" "0 0 0"
"speed" "100"
"wait" "4"
"targetname" "door1"
{
...
Additional information on colours and textures
...
}
}
```

Table 4.4 – Some Hammer MAP Code

O2.4 Configurations should be in an interchangeable\exchangeable format

As outlined in the previous chapter, an ontology-based approach has been proposed for creating configuration policies. This means that an ontology format needs to be selected. For this system, the Web Ontology Language, OWL was chosen.

OWL was selected as it provides a good set of reasoning logic and allows for the encoding of features such as cardinality and object relationships that will be useful in this experiment. It also allows for the distributed storage of knowledge, which is very useful in a pervasive computing environment.

There are also a number of freely available Open Source tools for carrying out reasoning on OWL ontologies.

In order to transform the output of the map editor into an ontology, two steps are required. Firstly, a set of ontology classes needs to be written to cover the application. Secondly, the map file needs to be transformed in this ontology.

The Protégé programme developed at Stanford was used to author the ontologies. The version used, 3.0 beta, is the first to provide dedicated support for OWL, and as it is open-source all of its backend code is available for re-use in other parts of this system.

The ontology itself is an extension of the Standard Ontology for Ubiquitous and Pervasive Applications, which is a semantic web proposal for applying their techniques to pervasive applications.

The hierarchy of classes in the ontology is shown in Figure 4.10. The parent Device ontology asserts a number of conditions and these are shown in Table 4.5.

Property	Condition
hasAllowedUsers	There must be at least one; must be an instance of the 'friend of a friend (FOAF) ontology'[4]
hasLocation	Must be only one location; must be an instance of the SOUPA geo-measurement ontology

Table 4.5 – Conditions Asserted by the Device Ontology

The subclasses then enforce their own particular conditions, such as Telephone must have one phone number.

A full copy of the ontology used is included in Appendix V.

[4] http://xmlns.com/foaf/0.1/#%22

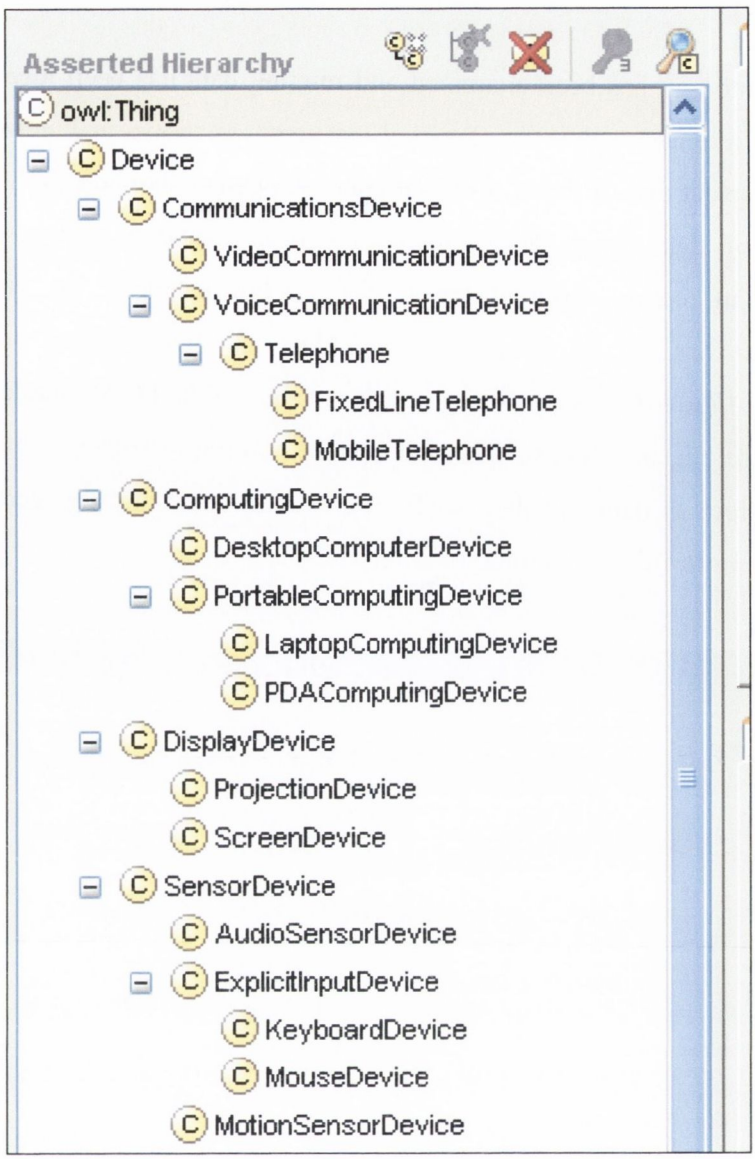


Figure 4.10 – The Device Ontology and Subclasses

O2.5 Constraints should be used to ensure models are sufficient and complete

The resulting OWL ontology is almost certainly incomplete. This is because the map editor only allows certain data to be captured and so the user must now be prompted to supply the rest. This means that the system needs to examine the ontology in order to discover what further data is required. In order to do this, the Protégé-OWL Plugin build 227 was used. A number of reasoning tools are available, however many of them make open-world assumptions about the target ontology. Such an assumption is not appropriate for this task and so the closed-world approach of this plugin is more relevant. The difference between closed- and open-world reasoning is discussed in Chapter 3.

When the ontology has been examined and missing data has been identified, the user is presented with a list of partial instances, as shown in Figure 4.11. When they select one of these, a tabbed panel is opened showing the various properties for that device, and allowing the user to supply values. These properties can require different types of data. For example, the property *hasPhoneNumber* can be satisfied by supplying an integer phone number, whereas *hasAllowedUsers* needs to given an instance of the FOAF ontology. In the latter case, the user is presented with a list of existing instances from which they can select as many as they wish. The system automatically recognises whether a simple value is required or an instance.

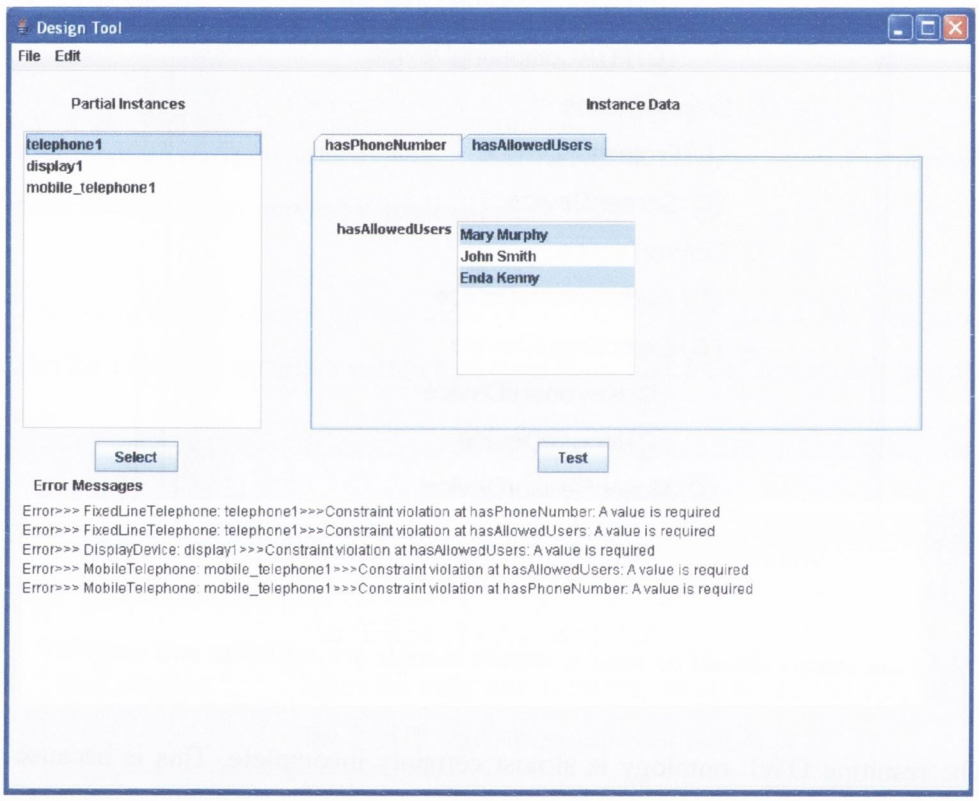


Figure 4.11 – Environment Modelling Tool with partial model

By adopting a convention that properties and classes have meaningful names, it is possible to use them as part of a dialogue with the user, which aims to fully capture required data. For example, by using the *hasPhoneNumber* attribute in a class *LandlineTelephone*, the system can automatically generate requests for this value that take the form:

Constraint Violation: LandlineTelephone hasPhoneNumber must have a value

This message is readily understandable and doesn't require complex language processing or AI techniques to make its point.

When the user has supplied the necessary information, they can request that the ontology be released to the environment. The system carries out a check before releasing the configuration, and if it still has insufficient information the list of partial instances is refreshed to show only those outstanding devices and the user will again be prompted to supply the missing data. When all device configurations are judged to be semantically complete, the policy is deployed.



Figure 4.12 – Map confirmation dialog

4.8.1. Limitations

This approach has a number of limitations. The main one being that by using a proprietary map editor, the system isn't fully integrated. Instead when one wishes to use the map editor, the main programme spawns a separate process. The main programme then blocks until Hammer quits and control returns to the parent. In order to give a sense of integration, the parent programme maintains two timestamps: one for when the Hammer process is created and one for when

it quits. When the user closes Hammer, the parent scans the map folder to find files modified between the two times. Any files that have been modified are presented to the user in the dialog shown in Figure 4.12.

However, the system is written in a very modular fashion, and so if a suitable alternative map editor was identified, it could be integrated with a minimum of fuss; the sole requirement being a new parser.

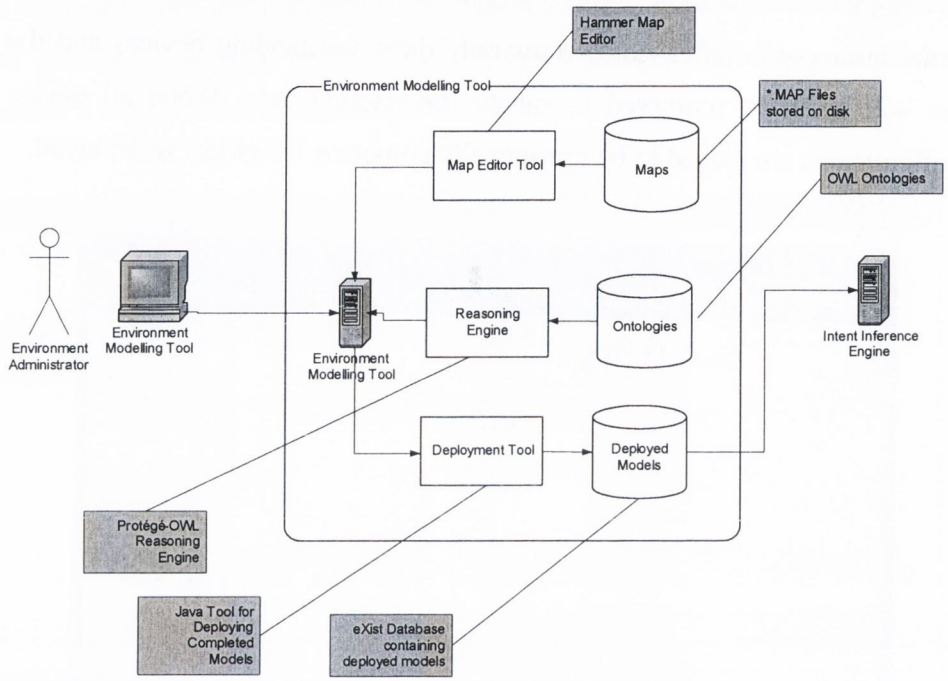


Figure 4.13 – Environment Modelling Tool and Implementation Technologies

Another limitation is that some level of expert knowledge is required to author the ontology classes that are then used to create the models. However, we assume that authoring these device ontologies is beyond the scope of our target users, and so it is therefore sufficient to provide tools that allow them to intuitively use the ontologies.

4.8.2. Summary

Figure 4.13 summarises the proposed implementation of the Environment Modelling Tool, and shows the chosen technologies.

4.9. ROUTINE BEHAVIOUR DESCRIPTION TOOL

4.9.1. Overview

In the previous chapter a number of design requirements were defined for the Routine Behaviour Description Tool, RBDT. These were:

O3.1 Provide a mechanism for describing routine behaviour
O3.2 Facilitate the production of a directed acyclic graph linking activities (nodes) and their causal inter-relationships (edges)
O3.3 Provide a requirements capture tool that minimises the cognitive load on the expert
O3.4 Identification of a suitable representation format
O3.5 Provision for the re-use of pre-existing activity fragments

In addition, a Fragment Definition Tool was provided for System Experts. For this implementation, JavaBayes itself was used as this tool.

O3.1 Provide a mechanism for describing routine behaviour

As covered in previous chapters, every human activity is a combination of a number of simpler contributory acts. Taken individually these acts could lead to any number of potential outcomes, however when taken together, they aggregate to produce a high-level outcome. In order to allow for the system to carry out inference, it is therefore necessary for it to understand the relationship between acts and also for there to be some method of interpreting the actions as observed by the sensors and context information available.

The proposed method for describing the causal relationships between concepts is a directed graph where the arcs between nodes describe their relationship. The state at any given node can therefore be described as some function of the input data from antecedent nodes. In turn the state of that node can inform the state of descendant nodes.

In the approach implemented here, the most basic nodes map to given sensor inputs. Combinations of these simple sensor inputs aggregate to human intelligible concepts or behaviours, such as 'audience seated'. These higher-level concepts can then be linked to potential system interventions.

O3.2 Facilitate the production of a directed acyclic graph linking activities (nodes) and their causal inter-relationships (edges)

In the directed acyclic graph, links exist between concepts indicating that the state of a given node can influence or inform the state of another, or lead to predictions about outcomes based on observed knowledge. Typically the creation of such graphs poses two problems. Firstly there is a need to create an initial model complete with data describing all nodes and the links between them. The second step involves training this network to improve efficiency or accuracy. A number of algorithmic and mathematical techniques exist for carrying out such training, and these can be used to produce the optimal graph for a particular well-defined problem – e.g. associating defined meteorological events, and then training these associations so that the resulting graph can be used to accurately predict weather conditions.

The approach discussed here focuses on the first challenge, namely programming the network in an intuitive way. As is discussed below, this involves abstracting certain tasks and replacing them with weights and a transformation algorithm. There is some comment in the evaluation chapter on how such networks might be trained in an ideal solution, but it is not provided within this prototype's implementation.

In these graphs, users must mathematically describe the influence or the cumulative effect of a given set of nodes on a descendent. This is not an intuitive proposition for a non-expert user, as it requires manually quantifying various states and providing discrete values for all potential outcomes.

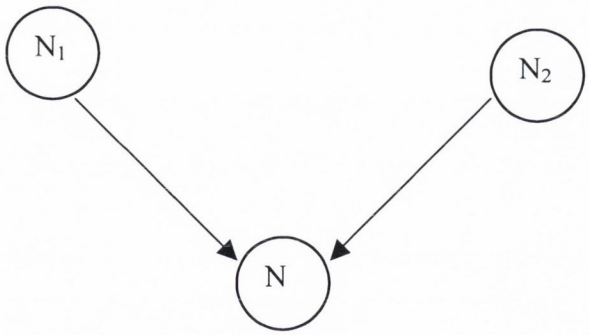


Figure 4.14 – A Simple Arrangement of Nodes

In order to overcome this, a simpler view of the relationships is given to the user. Consider the arrangement in Figure 4.14. Input nodes N_1 and N_2 can either be in a true state or a false state. The state at N is therefore dependent on N_1 and N_2. If N_1 and N_2 both have equal influence on N then the utility function U at N would be as shown.

$$U(N) = f(N_1, N_2) = \frac{U(N_1) + U(N_2)}{2}$$

However, if N_1 and N_2 have unequal influence, then some weighting will be necessary in order to overcome this. This gives an equation of the form:

$$U(N) = f(N_1, N_2) = \frac{w_1 N_1 + w_2 N_2}{w_1 + w_2}$$

This function is a specific case of the general weighted average function:

$$\overline{N} = \frac{\sum_{i=0}^{n} w_i N_i}{\sum_{i=0}^{n} w_i}$$

If the weight at a given node is viewed as a measure of its relative influence on the outcome at the child, then this function can be used to calculate the value of the child assuming the weights are known.

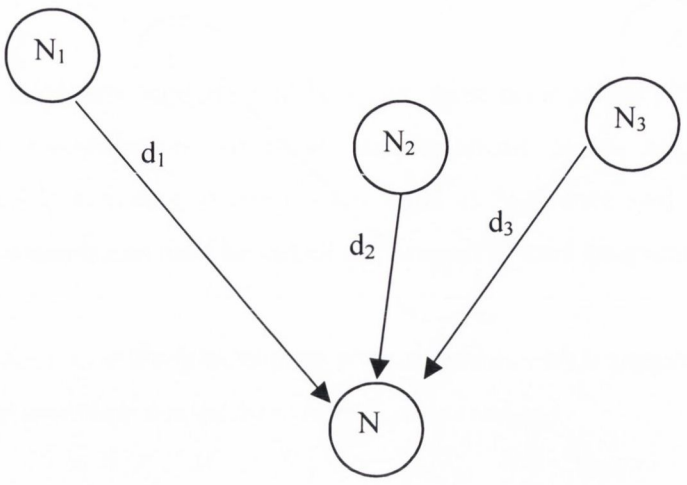


Figure 4.15 – A Network of 3 Input Nodes

Consider the set of nodes in Figure 4.15. Say the weight at each node was equal to its relative distance from the child node N. This means that the weight at a given node is:

$$w_n = d_n$$

However, weights calculated in this way increase as the distance increases. In the system proposed here, the weight should increase as the relative distance between a given node and N decreases.

Relative distances can be calculated as a fraction of the overall distance. This produces the following expression:

$$w_n = \frac{d_n}{\sum_{i=0}^{j} d_i}$$

This states that the weight is a fraction of the distance at n over the sum of all the distances of the nodes in the system. However, this again increases with d_n. However, if one instead considers:

$$w_n = 1 - \frac{d_n}{\sum_{i=0}^{j} d_i}$$

then the relative weight increases as d_n decreases. This can be rewritten as:

$$w_n = \frac{\sum_{i=0}^{j} d_i - d_n}{\sum_{i=0}^{j} d_i}$$

However, returning to the general form of the weighted average

$$\overline{N} = \frac{\sum_{i=0}^{n} w_i N_i}{\sum_{i=0}^{n} w_i}$$

it can be deduced that since $\sum_{i=0}^{n} w_i$ can be expressed as:

$$\sum_{i=0}^{n} w_i = \sum_{j=0}^{k} \frac{\sum_{i=0}^{j} d_i - d_j}{\sum_{i=0}^{j} d_i}$$

This in turn can be rewritten as:

$$\sum_{j=0}^{k} w_j = \sum_{j=0}^{k} \left(1 - \frac{d_j}{\sum_{i=0}^{n} d_i} \right) = k - \frac{\sum_{j=0}^{k} d_j}{\sum_{i=0}^{n} d_i}$$

Since $k = n$ as they both represent the maximum number of nodes in the system, the sum of the weights therefore becomes:

$$\sum_{j=0}^{k} w_j = k - 1$$

This means that the utility function at a given node, where the weights are equivalent to the relative closeness of each node becomes:

$$\overline{N} = \frac{\sum_{i=0}^{n} w_i N_i}{n-1}$$

where the weight at each node can be computed as:

$$w_n = \frac{\sum_{i=0}^{j} d_i - d_n}{\sum_{i=0}^{j} d_i}$$

However, a number of outcomes are possible at each node and each will have its own utility function. Consider again a 2 nodes system with a utility at N that depends on the states of these two input nodes. The outcomes can be summarised as in Table 4.6.

Input States		Utility Function
N_1 = true	N_2 = true	$f(N_1,N_2)$
N_1 = true	N_2 = false	$f(N_1,\neg N_2)$
N_1 = false	N_2 = true	$f(\neg N_1,N_2)$
N_1 = false	N_2 = false	$f(\neg N_1,\neg N_2)$

Table 4.6 – States and Utilities in a 2 Node System

For more than two cells there will be 2^n combinations of inputs for n input nodes. In order to compute the utility values for every combination at a given node, without knowing in advance the number of input nodes requires an algorithmic approach.

The approach taken here is to visualise the progression through combinations as a binary count. Taking again the system in Table 4.6 for a two-node table,

this can be rewritten as a binary count where trues are zeroes and ones represent falses the result is:

Input States		Utility Function
N_1	N_2	
0	0	$f(N_1,N_2)$
0	1	$f(N_1,\neg N_2)$
1	0	$f(\neg N_1,N_2)$
1	1	$f(\neg N_1,\neg N_2)$

Table 4.7 – Binary Representation of a Two-Node System

This would lead to four general utility functions at N, namely:

$$U(N) = f(N_1, N_2) = \frac{w_1 U(N_1) + w_2 U(N_2)}{w_1 + w_2}$$

$$U(N) = f(N_1, \neg N_2) = \frac{w_1 U(N_1) + w_2 U(\neg N_2)}{w_1 + w_2}$$

$$U(N) = f(\neg N_1, N_2) = \frac{w_1 U(\neg N_1) + w_2 U(N_2)}{w_1 + w_2}$$

$$U(N) = f(\neg N_1, \neg N_2) = \frac{w_1 U(\neg N_1) + w_2 U(\neg N_2)}{w_1 + w_2}$$

If all of the true input utilities for k nodes $U(N_k)$ are computed and placed in an array, and each $1-U(N_k)$ representing the false utilities is placed in a second array, a loop can then be used to cycle through the 2^n output states.

Table 4.8 illustrates how such a count would occur for $k = 3$ states. It is followed by an exploration of how a set of utility values can be constructed for such an arrangement. This examination will be used to deduce a general case for any number of input nodes.

Cycle Count	Binary Count
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Table 4.8 – A Binary Count in a 3 Node System

On each iteration, the binary encoding of that cycle position can be used to decide whether to choose a value from the trues array or the falses array in order to compute that utility function.

This results in the following utility functions:

$$U(N) = f(N_1, N_2, N_3) = \frac{w_1U(N_1) + w_2U(N_2) + w_3U(N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(N_1, N_2, \neg N_3) = \frac{w_1U(N_1) + w_2U(N_2) + w_3U(\neg N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(N_1, \neg N_2, N_3) = \frac{w_1U(N_1) + w_2U(\neg N_2) + w_3U(N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(N_1, \neg N_2, \neg N_3) = \frac{w_1U(N_1) + w_2U(\neg N_2) + w_3U(\neg N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(\neg N_1, N_2, N_3) = \frac{w_1U(\neg N_1) + w_2U(N_2) + w_3U(N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(\neg N_1, N_2, \neg N_3) = \frac{w_1U(\neg N_1) + w_2U(N_2) + w_3U(\neg N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(\neg N_1, \neg N_2, N_3) = \frac{w_1U(\neg N_1) + w_2U(\neg N_2) + w_3U(N_3)}{w_1 + w_2 + w_3}$$

$$U(N) = f(\neg N_1, \neg N_2, \neg N_3) = \frac{w_1U(\neg N_1) + w_2U(\neg N_2) + w_3U(\neg N_3)}{w_1 + w_2 + w_3}$$

Overall, by combining the utility function with the method described above for choosing appropriately from either the true or false value of each input node, one can generate a complete set of values for the range of outcomes and input states at a given node with any number of potential input nodes. To programme such a network, the user need only work with the relative distances and thresholds at each node.

O3.3 Provide a requirements capture tool that minimises the cognitive load on the expert

[Lenat, 1990] states that graphical interfaces provide an intuitive means of allowing users to work with complex knowledge. The approach proposed here is therefore a graphical one. Users are presented with a pane into which they can draw individual nodes. These nodes are assigned names and the user can then draw links between the nodes. This graphical component was developed using the JGraph API tool and is shown in Figure 4.16.

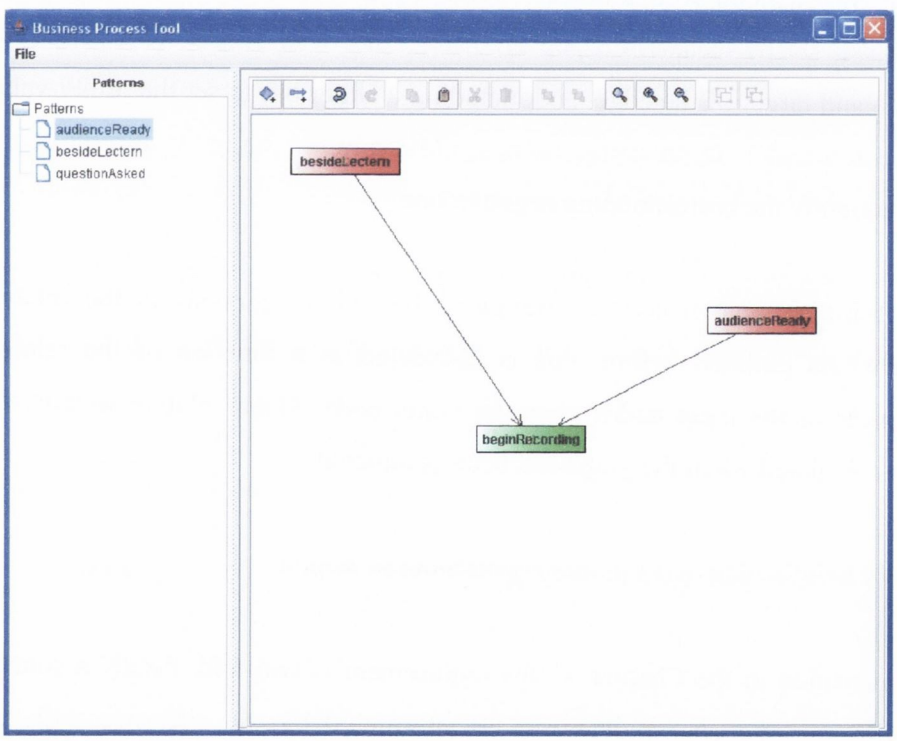


Figure 4.16 – Routine Behaviour Description Tool

In JGraph it is possible to assign 'User Objects' to cells and edges, the JGraph notation for nodes and the arcs between them. Such user objects were defined to store utility information on the edges.

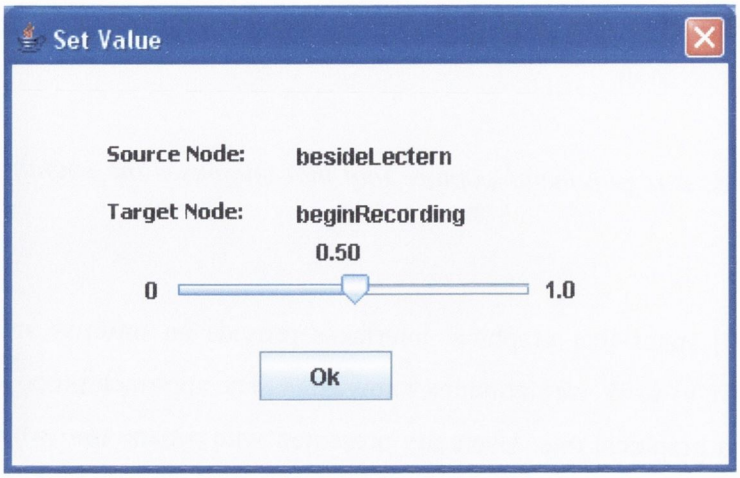


Figure 4.17 – Threshold Popup Window

These BayesianGraphEdge objects stored the utility value for that edge. This value was set by means of a pop-up dialog, shown in Figure 4.17, listing the source and target of the edge, and including a slider bar to set the utility value between 0 and 1. In so doing, users could easily set the utility threshold and consequently the corresponding negative one.

The third variable needed to complete the utility functions is the relative weight. As outlined before, this is calculated as a function of the relative distances of the input nodes from the target node. These relative weights are only calculated when the graph has been completed.

O3.4 Identification of a suitable representation format

As identified in the Chapter 4, this requirement is two-fold. Firstly a suitable format must be chosen that can be interpreted by the inference tool, and secondly the configurations need to be stored in a network accessible, stable, reliable and editable location.

Taking the format challenge first, the inference tool is based on the JavaBayes API and so it would seem reasonable to choose one of the file formats supported by JavaBayes. In addition, as the majority of data used elsewhere is in XML, it would seem prudent to go for the XML BIF format as this may enable the re-use of existing code. The Document Type Definition (DTD) for XML BIF is shown in Table 4.9.

```
<!DOCTYPE BIF [
    <!ELEMENT BIF ( NETWORK )*>
    <!ELEMENT PROPERTY (#PCDATA)>
    <!ELEMENT TYPE (#PCDATA)>
    <!ELEMENT VALUE (#PCDATA)>
    <!ELEMENT NAME (#PCDATA)>
    <!ELEMENT NETWORK
        ( NAME, ( PROPERTY | VARIABLE |
PROBABILITY )* )>
    <!ELEMENT VARIABLE ( NAME, TYPE, ( VALUE |
PROPERTY )* ) >
    <!ELEMENT PROBABILITY
        ( FOR | GIVEN | TABLE | ENTRY |
DEFAULT | PROPERTY )* >
    <!ELEMENT FOR (#PCDATA)>
    <!ELEMENT GIVEN (#PCDATA)>
    <!ELEMENT TABLE (#PCDATA)>
    <!ELEMENT DEFAULT (TABLE)>
    <!ELEMENT ENTRY ( VALUE* , TABLE )>
]>
```

Table 4.9 – XML BIF Document Type Definition

Taking the XML BIF format, data is stored in two ways. There are 'variables' which correspond to the nodes in the graph, and 'definitions' which define the potential outcomes at each node.

The XML BIF representation of Table 4.10 is shown in Table 4.11.

This configuration can be created by analysing the graph. Firstly individual nodes are identified and used to create the variable nodes. The edges in the graph are then identified and using the approach outlined in the last section the utility values for each are calculated and along with their inverses used to create the definition nodes.

Utility Function		Inverse	
$f(N_1,N_2)$	0.8	$f'(N_1,N_2)$	0.2
$f(N_1,\neg N_2)$	0.65	$f'(N_1,\neg N_2)$	0.35
$f(\neg N_1,N_2)$	0.3	$f'(\neg N_1,N_2)$	0.7
$f(\neg N_1,\neg N_2)$	0.25	$f'(\neg N_1,\neg N_2)$	0.75

Table 4.9 - Sample Marginal Utility Values

Now that the configuration policy has been prepared, it needs to be deployed. In order to fulfil the availability and reliability demands, it was decided that the configurations should be deployed to a network accessible database. A number of XML databases are available, as discussed in the State of the Art, and eXist was chosen as it has the best support for XPath and XQuery as well as providing a number of stored procedures that allow easy interaction with the database from Java.

In the database, a collection was created called 'deploy' to hold the configurations. When the user is satisfied with their configuration, they are presented with a pop-up window, see Figure 4.18, which allows them to select a date and venue for the configuration. The configuration is then deployed to the database as follows:

```
/db/deploy/[month]/[day]/[time]/[venue]/config.xml
```

In so doing, only one configuration is permitted for a given combination of date and venue, as previous configurations at that combination of date and venue will be overwritten.

```
<bif>
<name>Example</name>
<network>
<variable>
            <name> N1</name>
            <type>discrete</type>
            <value>true</value>
            <value>false</value>
</variable>

<variable>
            <name> N2</name>
            <type>discrete</type>
            <value>true</value>
            <value>false</value>
</variable>
<variable>
            <name>N</name>
            <type>discrete</type>
            <value>true</value>
            <value>false</value>
</variable>
<definition>
      <for>N</for>
      <given> N1 </given>
      <given> N2 </given>
      <table>0.8  0.65  0.3  0.25  0.2  0.35  0.7  0.75
</table>
</definition>
</network>
</bif>
```

Table 4.11 – the XML BIF Version of Table 4.10

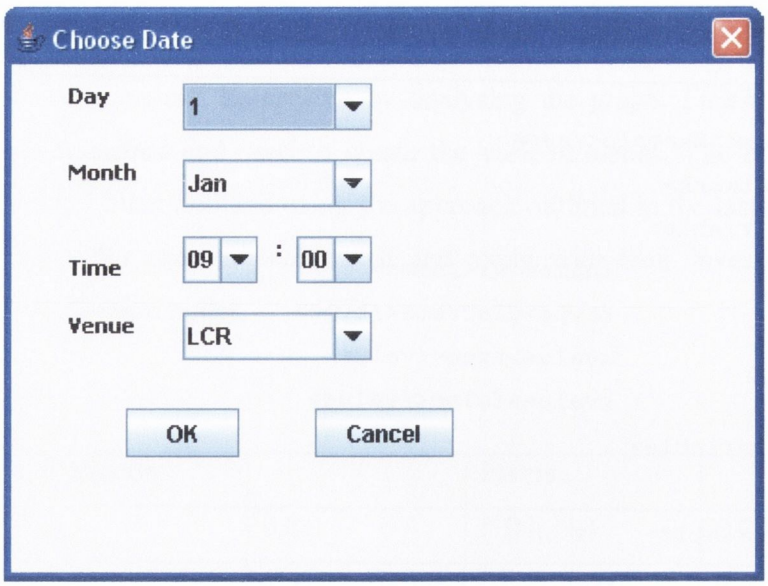


Figure 4.18 – The Deploy Popup

O3.5 Provision for the re-use of pre-existing activity fragments

Activity fragments range from the basic human level concepts such as *besideLectern* through to entire configurations. It is important that these fragments are available so that they can be reused. This is done by creating an eXist XML pattern repository where fragments are grouped according to types of activity. The grouping policy would ultimately depend on the types of typical activity that take place in the space, which in a business for example might mean grouping around Meetings, Seminars, Negotiations and Presentations. Users can import these fragments into the current model and then when they are ready to deploy the system automatically imports the XML fragments corresponding to each of the imports and appends them to the current document.

4.9.2. Summary

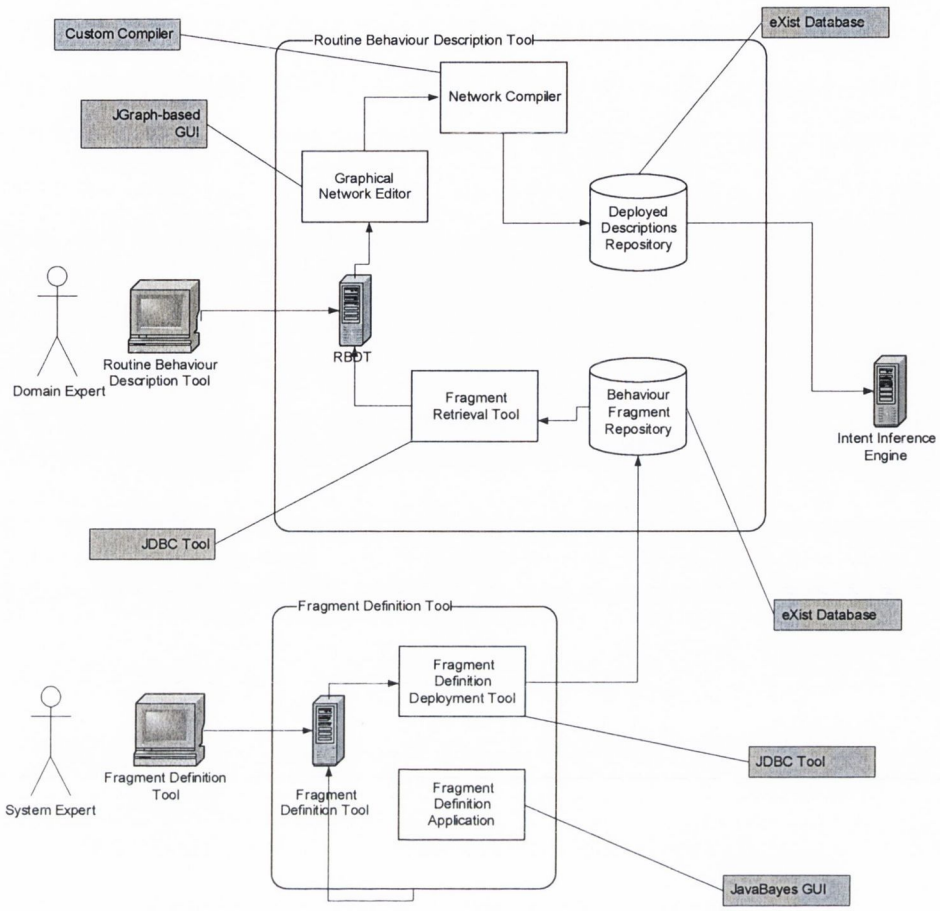


Figure 4.19 – RBDT with Implementation Technologies

Figure 4.19 shows the RBDT architecture including the chosen implementation technologies.

4.10. OVERALL ARCHITECTURE

Now that the system has been presented in each of its subcomponents, it would be worthwhile to return to the process model presented at the start of this chapter. Figure 4.20 shows the original architecture from the start of the chapter, but now it also includes details on the how the various components integrate together to collectively support the ordinary users as they interact with the environment.

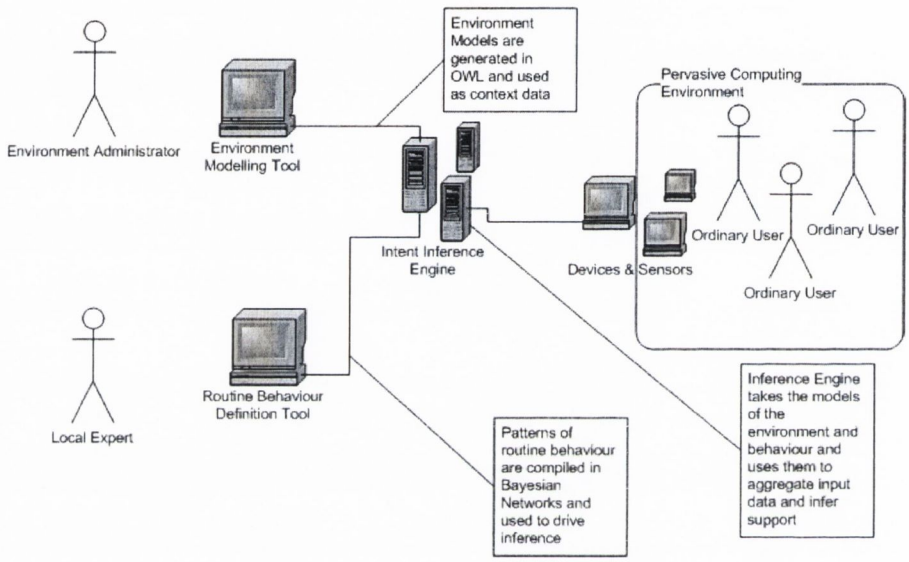


Figure 4.20 – Overall Governance Architecture

The Environment Administrator and Domain Expert roles are now supported by components that allow them to model the environment for context purposes and describe behaviour in a structured way. The Intent Inference engine can then operate on the outputs of these to respond to user behaviour as detected by sensors in the Pervasive Computing Environment. It can then make inferences about the users intentions and therefore suggest support in an appropriate and accurate manner.

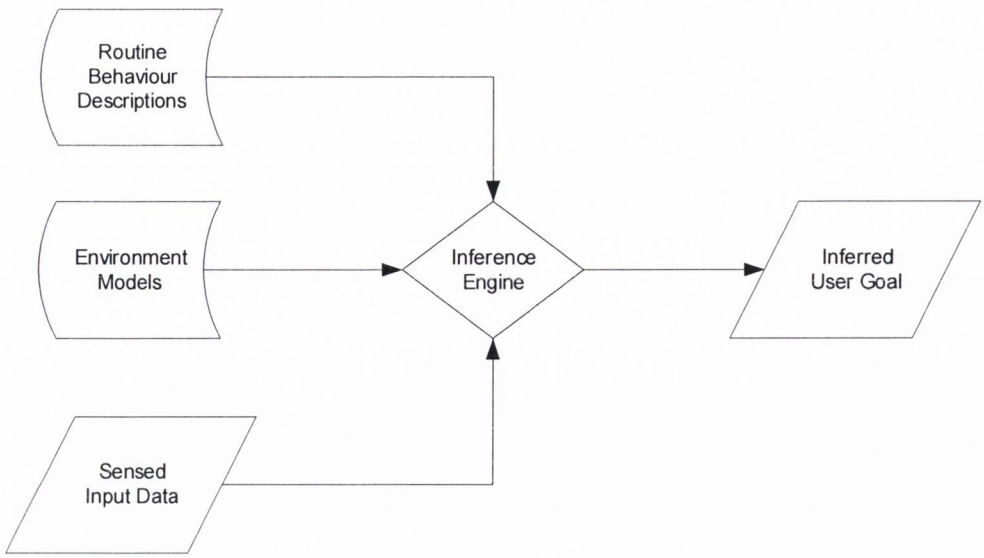


Figure 4.21 – Interaction of Information Flows

Figure 4.21 provides a simplified view of how the information flows between the components interact. Stored data from the RBDT and the Environment Modelling Tool is combined with dynamic data produced by sensors to make inference decisions and deduce the user's goals.

5. EVALUATION

5.1. OVERVIEW

This chapter presents an evaluation of the prototypes and overall objectives proposed in this thesis. The assessment is divided into a number of parts: sections 5.2 to 5.4 present an experimental evaluation of the three prototypes to assess how they met their design objectives; 5.5 provides a critical review of how the overall system meets the thesis objectives; and section 5.6 offers a comparative review of related work.

The overall integration of the system was tested by verifying that the output artefacts produced by the Environment Modelling Tool and the Routine Behaviour Description Tool could be successfully used by the Intent Inference Engine. However, no over-arching test was performed involving all three components in a single experiment.

Where the evaluations refer to test users, the individuals in question are postgraduates from the Computer Science Department of Trinity College Dublin. While technically literate, the selected individuals were not involved with the work presented here and they had little or no prior knowledge of the objectives or characteristics of the various components.

5.2. EVALUATION OF THE INTENT INFERENCE ENGINE

5.2.1. Experimental Objective & Means of Evaluation

The experimental objective is to evaluate whether the proposed approach to inferring user intent can successfully recommend appropriate support in a simulated environment, and whether this recommendation satisfies the test user. The assessment is based on two metrics namely: examining the outputs of the system; and recording user feedback via a questionnaire, see Appendix II.

The outputs of the system include notifications as evidence is added to the model as well as the final inferred recommendation of the system.

The questionnaire assesses the user's impressions of the system including a measure of their satisfaction. This is a useful source of data, as a successful pervasive computing environment must be both accurate in its inferences and also attentive to the expectations of the user.

5.2.2. Evaluating the Objectives

The prototype Intent Inference Engine examines the extent to which a Bayesian Network driven, mixed initiative system can address the thesis requirement to provide a means for aggregating and responding to sensed and context data in a pervasive computing environment.

Deriving from this, two design objectives were identified and these have been experimentally evaluated. These objectives are noted below.

O1.1 Correlate sensor and context data

O1.2 Perform utility-based inference on correlated data

O1.1 is focussed on assessing how accurately data produced from context and sensor sources can be correlated to provide input to the Bayesian Network. In the design proposed here, such correlation is an essential pre-processing step that allows heterogeneous inputs to be related to nodes in the Bayesian Network.

The assessment is based on examining the evidence notifications logged by the system in response to user activity and assessing how accurately they reflect the user's actual activities. An accurate correlation will result in the relevant Bayesian nodes being activated, ultimately contributing to correct inferences being made.

While O1.1 is focussed on producing useful inference input data, O1.2 addresses the component's ability to respond to such inputs and make accurate inferences on user's intentions. To experimentally meet this objective the system must successfully respond to users as they attempt a number of specified tasks. Each inference can be assessed by examining whether it accurately reflects the stated goal of each task. Successful inference will result in the system recommending the appropriate support.

In addition, the user feedback captured by the questionnaire also offers insights into the utility to the user of these recommendations. This can be gauged by assessing the user's satisfaction, which relates to their impressions of the usefulness of the system's pre-emptive interventions.

5.2.3. Experimental Approach

Seven test users with minimal prior knowledge of the system were selected and asked to attend an individual experiment session. Initially the subjects were given a verbal overview of the system objectives. This overview was at a task level, that is, it described the kind of interactions a user might have with a pervasive computing environment, and the type of pre-emptive support it could offer. For example, comments about a room being too warm might see the environment respond by opening a window or adjusting the air conditioning.

The users were then introduced to the simulated office environment, see Figure 4.5 in the previous chapter, and a brief demonstration was given of the three types of supported input: typed text to simulate speech; moving the mouse pointer to simulate gaze; and mouse dragging to simulate simple gestures.

Once the user was satisfied that they understood how to interact with the simulator, they were given three tasks to attempt. They were free to choose the tasks in any order and to interact with the simulator as they saw fit. These three tasks were to make a telephone call, transfer a file from one device to another and to transmit a file to a colleague.

As the simulator makes use of a limited vocabulary, the subjects were also given a dictionary of the words in this vocabulary. This dictionary is shown in Table 5.1. This approach is similar to the Lumière experiment [Horvitz, 1998] where a specific subset of actions in Microsoft Excel were specified to the user and the system then attempted to assist them as they performed these limited tasks.

Type	Values
Colleague's Names	Peter, Mary, Anne, Sean, Susan, Michael, Karen, Fred, Laura
Verbs	Send, Phone, Copy, Show, Call, Transfer, Display, Talk, Play, Ring, Move

Table 5.1 - Colleague Names & Verbs

In the case of filenames, the system uses the file extension as a source of context data. For example a printer device could not display an MPEG file representing moving images.

The users were free to attempt the tasks without further direction, and they were passively observed as they proceeded through the experiment.

Once all three tasks had been attempted and the results recorded, a post-experiment questionnaire, see Appendix II, was administered. This questionnaire enquires about user satisfaction and general feedback on their experience with the system.

5.2.4. Results

An objective-by-objective discussion of the experimental results is included below; however the key result of the experiment is that of the twenty-one individual tasks attempted by the test users, the system correctly inferred their intentions nineteen times. This demonstrates that there is merit in a Bayesian

Network driven approach to aggregation and inference. However, as the experiments involved a very limited set of potential inputs and outcomes, such a high level of success must be tempered by recognition of these limitations. These limitations are discussed in more detail below.

Among the more valuable feedback supplied via the questionnaires were comments on how the system behaved when it unsuccessfully inferred the user's needs. Some of the users concerned felt that the system should offer some qualified feedback rather than remaining inert, thus causing the user to re-express their request from afresh. Such an approach might involve a dialogue between the system and the user; however this would need to be sensitive to the importance of the task and the expectations of the user. If the system repeatedly interrupts the user's activities then there is a risk that it may be disabled and so such interruptions should be minimised.

One user felt that in general he preferred the traditional input modes, as they are 'tactile', except when he was attempting to make a phone call or other communication. In a real world implementation, this concern could be reflected by tailoring the system's view of each user by employing profile data. Ultimately, such concerns represent a need to meet a given users' expectations by recognising that while some individuals would appreciate a highly automated environment, others might prefer to retain a significant amount of control for themselves.

Another user felt that the system's responses could have been 'friendlier'. Again this is principally a reflection of the limited capabilities of the prototype. However, it does further indicate the need to meet each individual user's expectations of how support should be offered in a pervasive computing environment.

5.2.5. Discussion

The results indicate that a Bayesian Network driven approach to utility-based event aggregation can deduce successful outcomes. It also shows that this

depends on having input data that has been suitably prepared for addition to the network. Further discussion of the results is given below and centred on the individual design objectives.

O1.1 Correlate sensor & context data

The high level of successful inferences, c. 90%, demonstrates that pre-processing input streams to produce correlated input to the Bayesian Network is a valid and useful approach. When reviewing the notifications logged by the system as evidence, a number of issues of note arise.

For example, the key determinant in producing an inference seems to have been the users' text input. When taken alone such inputs' influence is limited by their associated utility, however the progress of all of the users involved preparing their 'state' before voicing their requirements. That is, the users used the mouse inputs first before moving onto the text input.

This could reflect a mode of real world behaviour, which would see a user arrange their resources first before actually taking their desired course of action. However, another potential explanation could be that since the mouse and keyboard both require the users' hands, they may feel more comfortable using them individually rather than at the same time. This leads to a situation where a user's first instinct when approaching a piece of graphical software is to use the mouse before other input modes.

On the correlations themselves, some resulted in the same node being fired more than once. For example one user gestured between the same two devices a number of times using the mouse dragging facility. This did not interfere with the system accurately identifying the devices concerned and inferring support.

The positive results do not reflect the difficulties faced if implementing such correlation in an accurate, efficient and timely manner with more complex inputs. The correlations carried out in this system were of limited complexity because the vocabulary and ontologies were entirely known in advance and therefore correlations could be defined explicitly and exhaustively in advance.

Such deliberate reduction in complexity has been used in order to test this type of system in the desktop domain [Horvitz, 1998].

One of the main challenges in developing more complete systems concerns determining the information content of evidence especially as this may degrade over time.

Another related concern is determining how to prioritise and filter input data in an environment where sensors are prolific. In the latter case, the utility-based approach could in itself offer a solution insofar as the utility of accepting information from certain sources could be reduced in order to attenuate its influence. This challenge has been identified by some related projects, and these are dealt with later in this chapter.

Overall, the correlation results indicate the benefits of ensuring that the inference engine is suitably prepared and informed before engaging in the inference process.

O1.2 Perform utility-based inference on correlated data

As data was added to the inference model, various competing outcomes were possible. However, based on the activities of the users, the system correctly inferred their intention in 90% of cases. Given the limited set of inputs and outcomes, a high-level of accuracy would be required for the approach to be deemed a success and so 19 out 21 is a solid result, but with significant caveats such as recognising that the experiment was based on a number of users attempting a single task at a time, with pre-defined vocabulary explicitly entered through the console and only a small number of potential types of support available.

However, these limitations have been explicitly recognised from the outset. Evaluation of the component's accuracy is therefore based on examining whether the core inference process can accurately make inferences. In this regard the system demonstrates that a Bayesian Network driven approach can

produce accurate utility-based inferences when user activity and context data are suitably correlated to provide appropriate input streams.

As noted previously, the users' feedback is also an important factor in evaluating the overall performance of the tool. Overall, satisfaction with the system was high with four users feeling 'Very Happy', two 'Happy' and one 'Neutral'. Of the two users who received incorrect inferences, one stated they were 'Happy', the other 'Neutral'.

In the case of the 'Neutral' user, they suggested that a more elegant approach to failure might be used in future. This could involve users being presented with a report of available evidence and allowing them to point out where errors had been made. This information could be used as feedback on the system or as a means of deciding what evidence to discard as the system continues to attempt to meet the user's needs. This level of detail may not be the preference of all users, and again this highlights the need for personalisation.

5.3. EVALUATION OF THE ENVIRONMENT MODELLING TOOL

5.3.1. Experimental Objective & Means of Evaluation

The need to model context intuitively is identified throughout this thesis as a key concern in enabling user governance of pervasive computing environments. The Environment Modelling Tool facilitates this requirement by allowing users to easily and interactively prepare models of local computing resources. The prototype evaluated here responds to the design objectives derived from the thesis objective.

In order to quantify whether these objectives were met, a number of metrics were employed including questionnaires, see Appendix III, a recorded user commentary and an evaluation of the output artefacts.

A multiple-choice questionnaire of forty statements formed part of the usability metrics. The questionnaire is based on the Software Usability Measurement

Inventory (SUMI) developed by the Human Factor's Research Group, at University College Cork [HFRG]. SUMI enables measurement of some of the user-orientated requirements expressed in the European Directive on Minimum Health and Safety Requirements for Work with Display Screen Equipment [90/270/EEC]. SUMI is also mentioned in the ISO-9241 standard as a recognised method of testing user satisfaction.

This questionnaire asks respondents to indicate whether they agree, disagree or are neutral with each statement. These statements are cross-referenced in order to provide sanity checking. For example, the statement "I enjoyed my session with this software" is cross-referenced with "There gave been times in using this software, when I have felt quite tense". Answering 'Agree' to both of these is inconsistent and so their relationship can be used to gauge the quality of the users' responses.

The other questionnaires asked about each user's prior experience of preparing a space for a task similar to the experiment, and there was also a post-experiment questionnaire. These are also included in Appendix III.

5.3.2. Design Objectives

Once again, the design objectives for this component are:

O2.1 System should produce a single environment model
O2.2 Users of varying skill should be supported
O2.3 Configurations should be in an interchangeable\ exchangeable format
O2.4 Constraints should be used to ensure models are sufficient and complete

These objectives are basically functional requirements of the system. Overall the guiding system requirement is that it should provide an intuitive method for producing complete and usable models.

5.3.3. Evaluation of Objectives

Each of the functional requirements is examined below, and this examination considers how each objective is to be assessed. In terms of the overall need to produce an intuitive and accurate modelling system; this can be evaluated by verifying the output models and reflecting on the feedback and commentary provided by each test user.

O2.1 System should produce a single environment model

The objective here is to produce one model of the environment that describes, in appropriate detail, each of the individual devices that are in the space. This single model should be sufficiently complete to act as a source of context data for the inference process, and if appropriate as a means of configuring the modelled devices.

This can be verified by examining the model produced by the user to ensure that it both contains the required devices, and that these devices are fully described. The first verification can be achieved by simply examining the model, while completeness can be checked using a standard reasoning tool.

O2.2 Users of varying skill should be supported

It is important that the tool should allow users of varying technical skill to exploit the ability to model devices. To examine this, users were given minimal training in the system so that their interactions are not based on prior experience. Their experimental interaction can be evaluated in two ways. Firstly, the user relates their usability experience of the system via the SUMI questionnaire. Secondly, by observing the user and following their commentary, deductions can be made about how closely their demands as expressed via the commentary matched their observed behaviour. This can give an indication of how efficient the system is at enabling the user and also how effective it is at providing a usable interface.

Taking these two means of evaluation together, conclusions can be drawn regarding how intuitively the system behaves.

O2.3 Configurations should be in an exchangeable format

The design calls for models to be in a format that can be shared among devices from heterogeneity of vendors so that they can be used for configuration. If one assumes that ontologies provide a means for sharing knowledge, then the question of producing an exchangeable format becomes instead a matter of producing an appropriate ontology. As has been outlined, the appropriateness or sufficiency of the produced models will depend on whether they fulfil their semantic constraints. This can be verified by ensuring that the output models are compliant OWL and that the data they represent is semantically complete.

O2.4 Constraints should be used to ensure models are sufficient and complete

The design suggests that by using appropriately defined constraints, the environment configuration tool can engage in a dialogue with the user to ensure that the models produced are sufficient and complete. This dialogue is driven by reasoning about constraints and using the output of this process to guide the user towards supplying additional required data. This dialogue continues until the reasoner concludes the models are semantically complete.

The effectiveness of this dialogue-based approach can be evaluated by assessing whether the user responds appropriately to the system's feedback, whether this feedback encourages a quick response and by examining the resulting models to check completeness.

5.3.4. Experimental Description

Six test users were selected with little prior knowledge of the system. Each user attended an experiment session individually. Before being introduced to the system, the users completed a questionnaire. This questionnaire asked whether

they had experience of preparing a room for a presentation-type scenario including preparing a room for a meeting, seminar or lecture. The types of preparation they were asked to comment on included configuration of local devices and providing access to the network and other resources for users in the space. The questionnaire asked the test subject to outline the steps they had to take in order to achieve this, as well as a rough estimate of the time taken.

Once the user had completed this questionnaire, they were introduced to the system. The introduction included a brief demonstration of how to use the Hammer map editor as well as a minimal overview of how to use the ontology tool. They were also shown where to find a previously prepared 3D map of the target space. Once they were satisfied with what was required, the demonstrator moved away and asked the subject to begin, reminding them to provide a verbal commentary as they went.

The demonstrator then observed the user as they made use of the system. If the user had any particular difficulties, they were encouraged to ask the demonstrator. These queries were included in the recorded commentary.

Upon successful completion of the task, the user was asked to complete a set of forty questions regarding their satisfaction with the system, as outlined above.

Finally, the users completed the second open-ended questionnaire where they could offer suggestions on future improvements to the system as well as general feedback on their experience.

5.3.5. Results

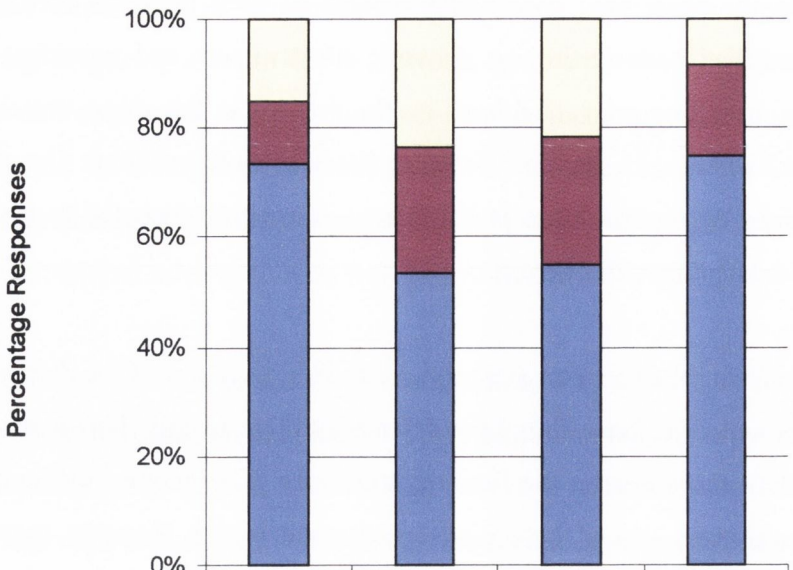
Each of the volunteers successfully created a model for the experimental scenario. On average, the experiments lasted 30 minutes, including questionnaires and demonstrations. This ranks favourably with the time test subjects said they spent when they attempted to complete similar tasks in the 'real world' where the average was 90 minutes.

Typically the most time consuming aspects of their previous configuration experience had been setting up network infrastructure and providing access accounts, installing relevant drivers on the devices in the space, creating user accounts, waiting for support from technicians and installing the required applications on the machines in the space; a properly designed configuration model could address all of these.

From analysing the user commentary and the observations of the demonstrator, the users appear to have found the system intuitive to use. This is also borne out by their responses on the final questionnaire. The greatest difficulty users experienced was around becoming accustomed to the Hammer map editor, however most quickly got used to the programme.

The multiple-choice responses were aggregated from forty statements into four main groups of responses which were then adjusted to take account of the context of the responses, for example disagree on a negative statement would correlate with agree on a positive statement. The results are shown in Figure 5.1. A complete account of the results on a per-question basis is included in Appendix III.

The first column covers a range of user satisfaction responses, such as whether the tool was pleasant to use and if they would recommend it to their colleagues. As can be seen their reactions were overwhelmingly positive with an aggregate satisfaction figure of 73%, and a minimal dissatisfaction of 15%.



	User Satisfaction	Reliability	Usability	Design Quality
Dissatisfied	15	23	22	8
Neutral	12	23	23	17
Satisfied	73	53	55	75

Figure 5.1 – Results of the SUMI Questionnaire

The second column shows users' opinions on reliability and how well the system responds to their interaction. The dissatisfaction rate here is higher at 25%. This is due to perceived 'clunkiness' in the map editor as well as some concerns about a lack of online help in the ontology tool. Users also expressed some dissatisfaction with the ontology deployment process, and specifically that once the system deems a model semantically complete it automatically deploys without allowing the user an opportunity to confirm. This approach means that the user can no longer add additional, non-necessary data once the model is judged sufficient.

The third set of results capture opinions on how easy it is to learn to make use of the system and whether it is intuitive or requires some change in user behaviour. The results here mirror the Reliability results and user opinions can be attributed to many of the same factors, especially the lack of an online help facility. It should again be noted that the demonstrator observed that some

users had particular difficulty getting accustomed to the Hammer editor, especially in trying to perform spatial reasoning in a simulated 3D world.

The last column reflects how user-centric the participants found the design. Again, the volunteers expressed positive opinions; however they did suggest some enhanced usability features such as drag and drop.

5.3.6. Discussion

All of the test subjects successfully completed the experiment, and more quickly than their previous experience with carrying out similar tasks in the real world. They also expressed satisfaction with the way system worked. Their feedback was generally positive, and satisfaction was high on each of the four criteria used by the SUMI questionnaire.

Taking each of the design objectives, they can be judged against the experimental results as follows.

O2.1 System should produce a single environment model

The approach proposed in this thesis is that devices of interest in the space can be modelled at an environment level, and that this process results in a single environment model that can be used as a source of context and configuration data. Each of the users was free to add whatever devices they felt would be required to the space and then they participated in a dialogue with the system to ensure the devices were sufficient as per the constraints defined in the ontology. Each of the users successfully created such models.

A potential concern for a real world implementation is the issue of how a desired model of context can be reconciled with the ultimate state of the environment when the planned activity takes place. This issue can be viewed as two distinct concerns: if the modelled devices ultimately are not present, how does the model deal with this; secondly if the devices are present, how does the context data become translated into configurations.

The first issue could be addressed by defining a protocol that would see devices in the space advertise their presence to the environment on their addition and that a register of these devices could be available to the inference engine.

The second concern could be addressed in a number of ways and would ultimately come down to a business policy decision within the enterprise that owns or maintains the environment. For example if devices need to be physically added to the space by the central administration, as opposed to their casual addition by the participants in the case of laptops say, then the system could use the model to issue work orders to maintenance staff. Another option would be for a more optimistic and laissez-faire approach that would assume that if the devices are present they adopt the configuration or privileges specified in the model. In cases where the device isn't present the solution of a register could detect this and update the model accordingly.

O2.2 The ability to produce models should not be unduly limited by user skill

By observing the user, it was noted that they found the design tool easy to use, although some users took a few minutes to become comfortable with the Hammer Map Editor. As designing in three dimensions requires some level of spatial aptitude, it would be expected that different users would have differing levels of initial success with the tool. However, even users who were initially uncomfortable quickly overcame this.

In addition, the usability results from the second questionnaire, especially under the categories of User Satisfaction and Design Quality, show that users found the system easy to use and also user centric. Both of these would be good indicators of intuitiveness.

On a user experience level, the need to interoperate with a proprietary third party product in a separate application is not ideal. If a suitable open source solution to replace Hammer could be identified and integrated into a single application this would help overcome this.

O2.3 Configurations should be in an interchangeable\exchangeable format

The resulting configurations were examined independently in Protégé to verify that they were semantically complete OWL ontologies. If one assumes that ontologies provide the sought after exchangeability, this should be sufficient for the objective.

O2.4 Constraints should be used to ensure models are sufficient and complete

Each of the users initially produced models that were not sufficient, and they were prompted to add the required missing data. In all cases, the users were successful, and they responded without fuss to the feedback given by the system via the constraints-based dialogue.

5.4. EVALUATION OF THE ROUTINE BEHAVIOUR DESCRIPTION TOOL

5.4.1. Experimental Objective & Means of Evaluation

The thesis proposes that the need to relate patterns of behaviour to preferred support outcomes is an important factor in user centric governance. The Routine Behaviour Description Tool provides a method to achieve this through the intuitive design and programming of Bayesian Network based models of behaviour. By limiting the level of explicit user input required, the RBDT attempts to address this need for intuitiveness. The author evaluated how the RBDT met the tools design objectives by creating models, verifying that they were valid and usable Bayesian Networks and by doing a comparison of the level of effort required in preparing similar models in the RBDT and the JavaBayes tool.

5.4.2. Design Requirements

The design requirements for the RBDT are defined in Chapters 4 and 5 as:

- O3.1 Provide a mechanism for describing routine behaviour
- O3.2 Facilitate the production of a directed acyclic graph linking activities (nodes) and their causal inter-relationships (edges)
- O3.3 Provide a requirements capture tool that minimises the cognitive load on the expert
- O3.4 Identification of a suitable representation format
- O3.5 Provision for the re-use of pre-existing activity fragments

5.4.3. Evaluating Design Requirements

Taking the evaluation of each of the design requirements in turn gives the following.

O3.1 Provide a mechanism for describing routine behaviour

The Intent Inference Engine examines how a Bayesian Network driven approach can be used to provide a utility-based mechanism for aggregating input data and making decisions based on knowledge about routine behaviour. The RBDT provides a method for capturing this knowledge.

This capability can be verified by taking the models produced by the RBDT and using them as inputs to the engine. This could be evaluated by developing a simulator similar to the testbed in the first experiment. However, as JavaBayes is the underlying inference platform, the models can be evaluated using the JavaBayes application itself. It allows users to load a network encoded in the XML-BIF format and to fire nodes in this network and observe results. The evaluation of the RBDT outputs can therefore be performed by loading these models into the JavaBayes application and supplying test evidence data to it.

O3.2 Facilitate the production of a directed acyclic graph linking activities (nodes) and their causal interrelationships (edges)

In order to produce a Bayesian Network, the RBDT needs to provide a method for designing a graph that can represent the nodes in the network and can store the data that each node encodes as well as the links between nodes. In terms of describing behaviour this representation will show activities as nodes with directed edges linking them to record causal relationships. This objective can be verified experimentally by taking a set of defined activities and their interrelationships and expressing them as a graph using the RBDT.

O3.3 Provide requirements capture tool that minimises the cognitive load on the expert

This objective will be evaluated by comparing the complexity involved in creating a Bayesian Network using both the RBDT and the JavaBayes application. As the models produced by the RBDT are being evaluated separately for accuracy, this comparison will instead focus on the level of explicit user input required to fully describe a Bayesian Network using each tool.

O3.4 Identification of a suitable representation format

Suitability in this case is a reflection of whether the output from the RBDT can be used as a basis for inference in the Intent Inference Engine. The engine makes use of JavaBayes as its underlying inference technology. It can accept a number of formats, but the one preferred here is the XML-BIF (Bayesian Interchange Format) file type. The evaluation involves assessing whether the graphical representation produced by the RBDT, including fragments of behaviour stored separately, can be combined and translated into XML-BIF and successfully loaded into the JavaBayes tool.

O3.5 Provision for the re-use of pre-existing activity fragments

The design calls for pre-existing patterns of behaviour to be made available to the domain expert. These patterns reflect any pre-defined combination of activity, and they are used as a means of binding user level activities to underlying sensor or other input. The evaluation will therefore consider the availability of these fragments, the ability to include them in a domain expert defined model of behaviour and the need to combine these fragments into a single output model for the scenario at hand.

5.4.4. Evaluation Approach

The evaluation of the RBDT differs from the other experiments, as it does not involve groups of test users completing a variety of tasks and then recording their opinions and outcomes. Instead the evaluation involved a scenario and describing the associated patters of behaviour in the RBDT. This behaviour was also described using the JavaBayes application. The evaluation involved both verifying that the model produced by the RBDT was a compliant and usable Bayesian Network and by examining the level of user knowledge and effort required when using both tools to complete a similar task.

The scenario selected for the evaluation is the Presentation Scenario covered in Chapter 2. In order to assist with defining the models of behaviour, some additional detail was added to that provided by the high-level scenario. This detail is shown in Table 5.3.

Scenario	Additional Information
Presentation	*Equipment\Sensors*: Multiple video\audio sensors, projector & screen *Policies* The lecturer stands at the front of the theatre Presenters do so from the lectern Presenters have control of the projector subject to intervention by the lecturer When a presentation is underway, the lights dim Audience members may ask questions by raising their arms Presentations are relayed over the Internet Presenters should wait their turn by standing at the front of the class Presentations last 10 minutes Non-verbal chat programs may be used by audience members Only the presenters voice will be broadcast, audience chat will not, save where an audience member is asking a question

Table 5.3 – Additional Information for RBDT Evaluation

In addition, some activity fragments were designed using JavaBayes and added to an eXist database. These fragments were available to the user and covered concerns such as 'arm raised' and 'standing at the front of the theatre'.

5.4.5. Results

A model representing the behaviour described in the presentation scenario was created using the RBDT and a similar model was developed using the JavaBayes tool. The RBDT model was successfully loaded in the JavaBayes tool. However, when viewed in the JavaBayes graphical tool, the nodes all

appeared at a single over-lapping point. This is because the XML-BIF file created by the RBDT does not store the Cartesian location of the various nodes. These co-ordinates are not used in any way by the reasoning functionality of the JavaBayes tool and are merely provided to facilitate network design in that tool.

Once loaded into JavaBayes, the nodes can be separated to make it easier to fire them and observe the impact that this has on neighbouring nodes. However, as the Intent Inference Engine does not require a graphical representation, this is not a limitation that would affect the successful operation of the overall system.

The level of explicit data demanded by the RBDT, and the associated need to think exhaustively and unambiguously, was considerably less than that experienced when using JavaBayes. Unlike the 2^n explicit user-supplied values required for n-input nodes and two output states in JavaBayes, the RBDT user need only supply 2n pieces of data. This data is given through the graphical slider and by the relative distances of the various input nodes. This means that the growth of required data through higher numbers of input nodes is far more scalable than the geometric progression provided by the JavaBayes approach, especially as it does not demand that a user consider explicitly the range of input and output states for every node.

It was seen that when developing the network using JavaBayes a waterfall style seems most effective. That is, the user draws the graph and then returns to each of the nodes to explicitly specify each of the 2^n values required for n-input nodes and two output states. The state of the network can then be tested and changes made as necessary.

In the RBDT case, the user was prompted to add the data to each node as they went, and while they could return to alter this data the lack of an in-application ability to test the state of the network was a limitation.

5.4.6. Discussion

Objectives O3.1, O3.2, O3.4 and O3.5 are relatively straightforward pass\fail concerns and given that they passed their associated test conditions, a further discussion is of limited value. However, O3.3 Provide a requirements capture tool that minimises the cognitive load on the expert, is worthy of discussion.

As noted in the results, section the level of user effort required to develop an initial network is considerably less than that required by the JavaBayes application. It also follows that an entirely manual process would be more demanding still. The RBDT approach also translates abstract numerical values into more human accessible concepts such as relative importance and thresholds.

Similarly the marginal increase in the level of explicit input data required by each additional input node follows an arithmetic progression in the RBDT as opposed to a geometric progression in the case of the JavaBayes or manual case. Given that the potential users of such a tool are domain experts rather than trained mathematicians this scalability offered by the RBDT algorithm is valuable.

While the inherent intuitiveness of the relative distance-driven weighting approach presented here has not been directly validated, it represents a significant reduction in task complexity when compared with other techniques for authoring networks such as using JavaBayes. As task complexity is a key determinant when assessing cognitive load [Crosby, 2003], the approach therefore reduces the load placed on the user.

However, a limitation in the RBDT arises when users want to move on from creating an initial network. This is because of two factors: there is no ability to both build and test within the RBDT; and there isn't an ability to directly alter the generated Bayesian Network, short of manually making changes to the resulting XML-BIF raw data. This reduced ability to manually fine tune is often a by-product of automation in systems: as interfaces are simplified the

ability to manually bypass the system is lost or hidden. As such, the trade-off is acceptable, especially considering the kind of users that have been identified.

This problem could be addressed by allowing the user to save their development model. This could be achieved relatively easily by serializing the object containing the map and saving this to disk, or as a Binary Large Object (BLOB) on a database. This could then be re-imported and edited in the RBDT. This would be a very valuable addition to the system, as a key motivation is to reduce cognitive load and this is currently frustrated by the difficulty in editing networks that have already been compiled and deployed.

In addition, since this component was evaluated, a plugin has been developed for JGraph that allows graphs to be saved as Scalar Vector Graphic (SVG) images. Again this could prove useful in sharing or saving graphical versions of the network.

The ability to build and test in a single environment is another desirable feature. At present, the RBDT has no capacity to interpret the model as anything other than a set of formatted data. This means that it cannot offer any ability to carry out testing and this has to be done in an external tool. Given that the Intent Inference Engine has already successfully exposed the required functionality, an improved RBDT might leverage this to provide a Bayesian Network Integrated Development Environment that would provide specific tools for developing utility-based models. This IDE would act as an alternative and less generic client to the existing JavaBayes component.

Finally, in Chapter 4 it is noted that for many Bayesian Network designers, the key challenge is how to take a defined network and train it to optimise performance or accuracy. This typically means that a given network for a particular well-defined application can be refined in order to make it optimal for that specific application. The resulting models are expected to persist for a time, and so the investment in training is worthwhile over the life of the network.

In pervasive computing, using resources to perfect a network that may be relatively short-lived and potentially disposable may not be appropriate. The design here therefore does not examine training. However, an ideal system would certainly benefit from training. In the case of the RBDT, this training could focus on optimising the weighting function for a particular domain rather than training a single network for a single use.

Training in this case could involve developing an experimentally verifiable use case and then using the current RBDT to provide an initial set of data. This could then be trained such that its ability to reliably produce results could be evaluated and incorporated into a feedback loop. The aim of such a loop would be to produce a more finessed version of the weights which are applied based on the graph produced by the user.

5.5. CRITICAL REVIEW

5.5.1. Overview

The three components evaluated above represent supports for the three stakeholder groups defined in Chapter 1, namely the Ordinary User, the Environment Administrator and the Domain Expert. This thesis contends that the overall goal of user-centric governance of pervasive computing environments can be served by identifying the particular governance needs of different types of users and providing intuitive tools that allow them to best exercise such governance. The approach should result in an integrated governance paradigm, where the different stakeholder systems interact to provide an overall basis for such governance. This critical review therefore examines whether the design approach has succeeded in providing such an integrated and intuitive system. It comments briefly on how the prototypes might satisfy the scenarios outlined in Chapter 2. It also involves an examination of related work so that the results obtained here can be placed in the context of the overall state of the art in governing pervasive computing environments.

5.5.2. Integration

The end user of the system proposed here is the Ordinary User. They will ultimately be the target of whatever pre-emptive support the system offers. The other components provide governance of the inference system so that it can best meet their expectations and deliver appropriate and timely support. The system that supports the ordinary user is therefore the point at which convergence occurs for the various system elements. To analyse this integration therefore requires an examination of how the Environment Modelling Tool and Routine Behaviour Description Tool input to the Inference Tool.

As noted in Chapter 1, the entire design process presented in chapter 4 began by developing the Intent Inference Engine and the requirements of this component drove the development of the other tools. By placing the Intent Inference Engine as the driver for the other components, the system seeks to ensure a point of integration that can ensure that the overall system functions as an integrated whole. This was verified by taking the Environment Modelling Tool and Routine Behaviour Description Tool, and verifying that they could provide meaningful inputs to the Intent Inference Engine.

Taking the Routine Behaviour Description Tool first, the Intent Inference Engine requires a Bayesian Network that can map sensed and context data to high level outcomes. The RBDT must therefore provide a suitably formatted and sufficiently complete model that can then be used by the engine to guide inference. The output of the RBDT is an XML configuration file deployed to an eXist database. This file contains a combination of user-defined activity patterns as well as predefined fragments that map sensor input to user level concerns.

The location in the database corresponds to a unique date and location that represents the delivery target for the configuration policy. The XML format is based on the XML Bayesian Interchange Format used by JavaBayes without the node location coordinates that JavaBayes uses to represent the Bayesian

Network graphically. It was verified that this node location data is not required for inference.

The meaningful integration of the RBDT and Inference Engine was verified by loading the configurations created by the RBDT into JavaBayes. The result is a set of co-located nodes, each sited at the same position in the graphical representation. However, while graphically at the same point, the edges logically linking the nodes still exist, albeit with zero length. The system is still able to carry out inference as a result without having to do any additional processing. Indeed, once the network has been loaded the user can manipulate it just as they would if it were created entirely within JavaBayes. They can also simulate observed evidence and verify that the suggested outcomes are as expected.

This demonstrates that the JavaBayes programme cannot identify either a syntactic or semantic difference between the RBDT networks and those natively created with the JavaBayes editor. As the inference process is built on JavaBayes, it can successfully use the RBDT networks.

It is worth noting that a business policy would be required to handle the length of time for which a policy is effective. In the deploy tool for the RBDT, the user can select a date and a time down to a quarter hour interval. When a new policy is deployed to a location, it overwrites any existing configurations. However, no end point is given for the policy.

A possible approach would be for a policy to remain in force until a new configuration is discovered. That is that a policy deployed for Room 101 at 10am would remain in operation until the system finds the next policy deployed for Room 101. This could be at 10:15 at the earliest, or at any other time in the future. This would be reasonable, as the users of the space could be confident that it would remain prepared for them for as long as they needed, and even if it remained faithful to their needs after they finish, the space would be vacant until the next scheduled event and so therefore there would not be an issue with leaving the configuration in place.

The Environment Modelling Tool also produces models which can be used in the Intent Inference Engine. The simulated 2D environment used to experimentally examine the engine uses device models as a source of context. The models produced by the Environment Modelling tool meet the requirements of this component. This demonstrates that the two are semantically compatible.

The modelling tool also deploys its configurations to the same location in the eXist database as the RBDT, although to a different destination filename than the config.xml created by the RBDT. As described previously, it is assumed that some mechanism in the environment can distribute this policy or make it available to local devices. This environment level technology could then report back on whether devices had successfully implemented the desired configuration via their declarative interface. This information could then be used to set context data in the inference engine or its pre-processor. For example, if one of the activities modelled in the Bayesian Network required a display terminal, and none was available, this entire possibility could be immediately dismissed as unsupportable.

Again, this interaction between the openly available environment model and the inference engine shows an integrated approach to governance.

The system also maintains a consistent vocabulary and graphical approach to configuration that shows an integrated design philosophy. Overall the approach behaves as different elements of an overall, single system, which is linked via the information streams outlined in Chapter 1.

5.5.3. Intuitiveness

The governance approach proposed throughout this thesis seeks to be user-centric. This has been defined as an approach, which seeks to be intuitive by lowering the cognitive load placed on the user. As has been stated, this approach seeks to make the configuration and user interaction with the

environment available to users irrespective of their level of system expertise or prior experience.

Overall the experimental results discussed above demonstrate that the proposed approach meets this requirement, albeit under a different metric for each of the components.

The Inference Tool seeks to reduce cognitive load by facilitating input via a wide range of sensors and context services. If the users' activities are sufficiently well captured by available sources, the event aggregation approach proposed would allow users to make requests using whatever natural mode they find most comfortable. In an ideally modelled and prepared space, this would reduce the additional cognitive load to close to zero as the user would not need to alter their normal behaviour to access support.

The Environment Modelling tool has two principle approaches to being intuitive. The first is to reproduce the target environment in a 3-dimensional, virtual representation. In so doing, the users are given an accessible method for describing their requirements for the space and regions within it. The second approach is to use an ontological approach to drive a dialogue between the system and the user to ensure the preparation of sufficiently complete models. This is achieved by taking care when naming the modelled entities and using meaningful names to describe their properties. For example, the 'LandlineTelephone' model had a property 'hasPhoneNumber'. The reasoner can therefore direct the user to supply a phone number by prompting them with the message 'Violation at LandlineTelephone: hasPhoneNumber must have a value'. This message is meaningful and transparent to the user and so they can easily meet the requirement it describes. The usability statistics gained from this experiment clearly indicate that the test users found the component intuitive and easy to use.

The Routine Behaviour Definition Tool attempts to reduce the cognitive load placed on the users by simplifying the method by which users can input their knowledge and by allowing them to visualise interrelated activities in an accessible, graphical format. The RBDT reduces the concepts of utility and

thresholds to a simple mixture of relative distances and sliding scales, and then provides an algorithmic method for processing these in order to produce a set of values covering all of the potential combinations of input and output states for the nodes in the model. This approach is manifestly simpler than the approach used in other tools such as JavaBayes where for a given node with 2 output states, true and false, and n input nodes there are 2^n values to be directly set by the user. This RBDT approach greatly reduces the cognitive load, and while the approach does not facilitate the same kind of direct manipulation capability, it does make the whole process considerably easier to use.

Overall the three components have demonstrated that they meet the need for the governance approach to be user-centric and intuitive within the defined criteria.

5.5.4. Returning to the Scenarios

In Chapter 2, three types of pervasive computing scenarios were described: a strongly role-based scenario (university presentations); a collaboration between peers (team meeting); and a highly personalised single user interaction (kitchen). Each of these presented different types of requirements. This section will discuss how the prototype might be used to meet the needs of each scenario.

Figure 5.2 represents the typical interaction of three components. Each scenario will be reviewed in terms of this pattern.

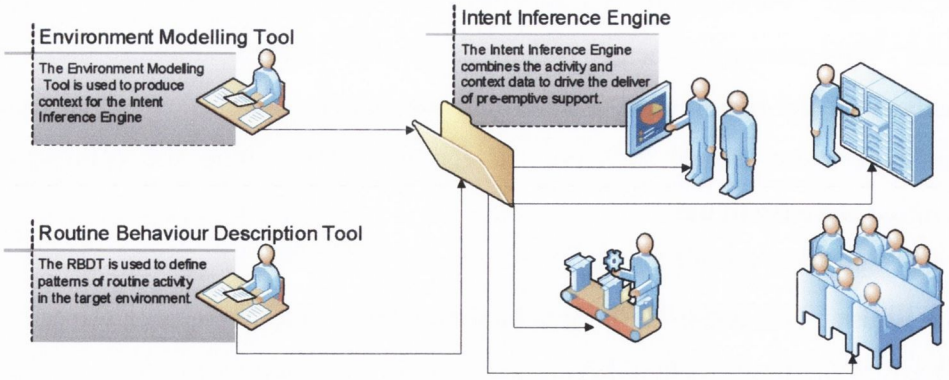


Figure 5.2 – The Integrated Governance System in Practice

Presentation Scenario

In the presentation scenario, a university lecturer is responsible for preparing a lecture theatre for a series of student presentations. In this scenario the roles in 5.2 would be filled as follows:

- **Domain Expert:** the lecturer
- **Environment Administrator:** the lecturer
- **Ordinary User:** the lecturer and students

The preparation process would involve the lecturer logging in to the RBDT and selecting a presentation-type behaviour definition. It would be expected that presentations are a pretty common and unchanging activity in a university, and so a suitable definition should be available. As a result, it is unlikely that she would feel the need to doing any further tailoring of the existing definition.

She would then access the environment configuration tool to ensure that the lecture theatre is adequately equipped. For example, the scenario identified a need to record the session and so the necessary audio\video equipment must be present.

Finally during the presentations, the Intent Inference Engine needs to track the behaviour of all of the different actors in the space who will have different types of interaction depending on their roles (audience, presenter or lecturer). It would also need to be sensitive to the changing roles that a given user may fulfil during the course of the session.

Meeting Scenario

In the meeting scenario, a design team are holding their weekly meeting. In this scenario, the user roles are less important as the various actors are effectively peers competing for the use of available resources. In this scenario, the system roles are exercised as follows:

- **Domain Expert:** the group leader
- **Environment Administrator:** the group leader
- **Ordinary User:** all team members

The environment administration tasks in this scenario are straightforward and require that the necessary resources are present. The Domain Expert role's responsibilities will depend on the size of the organisation. In a large organisation, it is likely that standard operating procedures will exist for most typical activities, such as team meetings. In this case, the group leader will merely need to select the most appropriate definition and assign it to their session. Alternatively, in smaller organisations or when the demands of the meeting are reasonably specific, the group leader will need to author a new definition or customise one that is pre-defined.

Finally, the group members will have equal access to resources, with the exception of the team leader who also has the power to revoke access in cases where the meeting is going over time. The Intent Inference Engine will therefore need to support this.

Kitchen Scenario

The kitchen scenario presents a domestic situation where there is a single user who does not want to face the burden of having to go through lengthy configurations. They would rather prepare their meal. This scenario also identifies an environment where the resources are unlikely to change very often, and all resources are always available and enabled to the user.

- **Domain Expert:** all patterns are predefined
- **Environment Administrator:** all resources are always available
- **Ordinary User:** the kitchen owner

The kitchen scenario explores a situation which is towards the automatic end of the governance spectrum. In this scenario, the user rarely expresses configuration demands and instead is entirely focussed on their daily tasks. The system therefore needs to know which type of activity they intend to attempt,

e.g. preparing a meal. Once this has been identified, and the corresponding behaviour description has been loaded, the system can monitor and support the kitchen owner. In time, it would be preferable for this system to learn about the owner's behaviour and improve the personalisation of the support offered. This would take the place of the more direct governance demands exercised by the domain experts and environment administrators in the other scenarios.

This scenario also identifies a need for devices to advertise their addition to the space, so that they become immediately available, as the scenario assumes that all resources will be available in all situations.

5.6. RELATED WORK

There are a number of cognate areas that the work presented here can be referenced against. A number are presented here.

5.6.1. Governance of Pervasive Computing Environments

The approach proposed here to user-centric governance of pervasive computing environments attempts to give an integrated paradigm that covers resource modelling, behavioural description support and intent inference. Other research projects are active in the area of management of pervasive computing environments, but they concentrate on particular aspects rather than addressing the entire problem space discussed here.

Research is underway at the University of Sussex on the creation of user policies for pervasive computing environments through natural language [Weeds, 2004] [Rimmer, 2005]. As with the work presented here, their project examines how inexpert users can express their demands on the resources in a pervasive computing environment. They also make use of ontologies as a means of overcoming the problem of distributing configuration requirements among heterogeneous devices.

However, their work differs in that they adopt a natural language approach while the work presented here works mainly on graphical interfaces. There is considerable work in the area of psychology on how different types of individuals deal with visual and verbal stimuli [Felder, 1988], and so their work (verbal) would be a compliment to the predominantly graphical (visual) approach proposed here.

Their research also differs in that it does not propose how the policies they create can be used to react to observed behaviour in the target environment.

[Christopoulou, 2005] presents a context-based reasoner for pervasive computing environments. It proposes a hierarchical approach to context using ontologies. Information generated from available context is used to assess the demands of the entity to which the context data refers, and this is then used to reason about what support should be offered to it. The reasoning process is based on a set of rules that are interpreted by the Java Expert System Shell, Jess [Friedman-Hill, 2003]. However, while the approach does address the inference challenge, it does not provide any intuitive method for generating the rules.

5.6.2. Mixed Initiative Interfaces

As discussed in Chapter 3, mixed initiative and attentive user interfaces have become popular in desktop computing, and they are now becoming a focus for interface designers in pervasive computing.

Eric Horvitz has been expanding mixed initiative and the underlying approaches to utility to more general issues of reasoning under uncertainty. From the perspective of this thesis, his work on the Microsoft S-SEER project [Oliver, 2005] is of particular relevance.

S-SEER provides a utility-based approach to classifying the value of data produced by sensors and context when trying to infer user intent. This value is then used to drive selective processing of data when considering outcomes.

This need to carry out inference based on multi-modal input is very similar to the need identified by the Intent Inference Engine, and the proposed solution is similar in its use of utility. However SEER uses hidden Markov models [Rabiner, 1993] as an underlying technology and focuses a great amount of detail on handling uncertainty. This differs from the motivation presented in this thesis, which is focused on providing event aggregation guided by easily captured user knowledge of routine behaviour.

His recent work has also included using Bayesian approaches to forecasting traffic flow in Seattle called JamBayes [Horvitz, 2005] and generating answers to free flow questions [Azari, 2004].

Other mixed initiative work in pervasive computing includes Confab [Hong, 2004]. It is a mixed initiative based toolkit for developing privacy-sensitive pervasive computing applications. Confab offers three levels of support: optimistic where applications share personal information and monitor for abuses, pessimistic where applications limit access to prevent abuses and mixed initiative where decisions are based on user-interaction.

However, while utility-based mixed initiative systems exist, the work presented here is novel in its close linkage to intuitively capturing user knowledge and using this to drive the pre-emptive delivery of support.

5.6.3. Interactive Ontology Creation

The Environment Configuration Tool demonstrates how well-formed ontologies can be used to enter into a dialogue with a user in order to create sufficient instances. A similar approach has been presented in the area of personalized learning [Denaux, 2005(1)]. It proposes a combination of an OWL ontology and a Jena reasoner to create a dialogue with users that can then be used to create personalized user models. The dialog is by way of a question and answer format and is based on the OWL-OLM application [Denaux, 2005(2)].

While this approach is focused on eLearning applications, in time it could provide a broader set of applications including the configuration of ubiquitous computing environments.

	Pervasive Computing	Intuitiveness	Bayesian Networks	Ontologies	Mixed Initiative	Interactive Ontology Creation	Context
This System	✔	✔	✔	✔	✔	✔	✔
Sussex	✔	✔		✔			✔
S-SEER	✔	✔			✔		✔
Confab	✔		✔		✔		
Denaux		✔		✔		✔	
JamBayes			✔				✔
Christopoulou	✔						✔
Beveridge		✔		✔			

Table 5.4 – Comparison of Related Work

Ontologies have also been used to create context specific interactions between a clinician and a system when diagnosing cancer [Beveridge, 2003]. However while the notion of a dialogue is similar to that presented here, it is not a constraints based approach to generating ontology instances. Instead it uses ontologies to respond to specific terminology used by the clinicians when answering questions posed by the system. Incidentally, this research also makes some use of mixed initiative.

While the use of ontologies to drive a question and answer dialogue has been employed in a number of other projects, the application presented in this thesis is novel as it focuses on easing the process of creating sufficient ontology models required by the system. This could be seen as using ontologies in a reverse from their usual direction insofar as they are being used to the system's benefit rather than that of the user. This reversal is novel as it enables the system to generate semantically complete ontology instances using a constraints based dialogue. Other applications focus more on using already existing ontologies to support users.

5.6.4. Overall Review

It has been shown that the strategies employed in the tools and approaches presented here are shared by other relevant research projects. As can be seen by the cited examples, the approaches described in this thesis are at least contemporaries and in some cases predate this other work. In addition the generic, user centric approach to managing pervasive computing presented in this thesis is novel and it provides a good addition to the other contributions.

A comparison of the features provided by the integrated system proposed here and the technologies discussed above is shown in Table 5.4. A number of specific areas are used for comparison, and each of these is a key component of the design proposed here.

This thesis argues that to be truly user-centric, systems must be intuitive and support the user through accessible and interactive tools. It adopts a Bayesian Network driven mixed initiative approach to satisfy users as they interact with a Pervasive Computing Environment. And finally the other three areas show how context can be modeled using ontologies, and how these ontologies can be created interactively.

The table demonstrates how the related projects draw on a combination of these topics, but none provides a solution that draws on the whole range.

Even limiting the comparison to just those projects that address Pervasive Computing demonstrates that the work presented here shares a commitment to intuitiveness, but the projects diverge in how this is delivered. These projects also identify context as a key requirement in Pervasive Computing, however only one, SUSSEX, also uses ontologies. However, SUSSEX does not provide an interactive way for producing them.

Overall the comparison demonstrates that the work presented here fits well within the current state of cognate research, however it also offers a novel alternative to the general state of the art.

6. CONCLUSIONS

6.1. INTRODUCTION

6.1.1. Overview

This thesis presents a novel approach to the challenge of governing pervasive computing environments. This governance challenge has been addressed for three groups of stakeholders in such environments, namely those who are responsible for modelling available resources, domain experts who have knowledge of routine behaviour and the ordinary users of the space. Each of these groups needs to be supported in order for the overall environment to be governed.

In addressing each group's needs, the objective has been to minimise the cognitive load on the users, and to make the system as intuitive as possible.

In this chapter, the objectives outlined in Chapter 1 are re-examined to assess how the work presented meets these objectives. This is followed by a discussion of this research's contribution to the state of the art in pervasive computing including a review of published papers. The chapter then concludes with a review of suggested future work.

6.2. THESIS QUESTION

This thesis assesses the extent to which human demands on the services and resources of a pervasive computing environment, and human knowledge on the routine modes of behaviour can be used to govern pervasive computing environments. This examination will focus on how human knowledge can be intuitively captured and used to programme Bayesian Network-driven, mixed initiative systems, and how sensed and context data can be used to drive the appropriate pre-emptive delivery of support to users.

6.3. OBJECTIVES

Three main objectives are identified in Chapter 1, namely to:

- **Investigate the governance requirements** for the flow of information between end-users and the system
- **Investigate and develop prototypes** that demonstrate an integrated approach to such governance
- **Perform experiments** to evaluate the proposed approach and prototypes

6.3.1. Investigation of Governance Requirements

Chapter 2 provides an investigation of the governance requirements of pervasive computing environments. It concludes that there are a number of types of relationships between such systems and their users, and it illustrates these relationships through case studies. These case studies focus on three scenarios: there is a clear leader and a number of relatively anonymous other participants (a college presentation); there are a number of peers competing for resources (a business project meeting); and a highly personalised single user example (preparing a meal).

It notes that there are some common demands such as the need to effectively model routine behaviour in a pervasive computing environment. It also identifies the specific requirements of each scenario such as the importance of roles in the presentation example, the need to support users across locations in the meeting and how domestic users may need to surrender some direct control to the system in order to reduce the overhead of interacting with the system on a daily basis.

The conclusions of this investigation are novel insofar as they attempt to make observations based on a number of very different applications. As shown elsewhere in this thesis, much current work focuses on narrow domains and so by adopting a generic approach, new insights can be drawn.

6.3.2. Investigate and Develop Prototypes

Chapter 3 presents a review of the state of the art of pervasive computing and concludes by identifying a number of technologies and approaches that form the basis for the design decisions and prototypes proposed in Chapter 4. The design approach focuses on supporting the three governance roles identified in Chapters 1 and 2, namely the ordinary users, domain experts and environment administrators. The design provides an individual component for each user, but each component interacts with the others to provide a single integrated platform for the delivery of governance.

The three prototype components that are defined provide a means for experimental evaluation of the design. Given the potential cost and complexity of developing a real world pervasive computing test harness, the prototypes instead make use of simulated 2-dimensional and 3-dimensional spaces.

The three prototype components are: the Intent Inference Engine which is a Bayesian Network driven application for aggregating evidence on context and behaviour and using this to infer user intent; the Environment Modelling Tool which allows users to model available resources so that they can be a source of context data; and the Routine Behaviour Description Tool which allows users to describe routine behaviour visually and then compile their descriptions into Bayesian Networks.

The integration of the components occurs at the Intent Inference Engine, which uses the outputs of the other components as input. The models produced by the Environment Modelling tool are used as a source of context data while the behaviour descriptions are used as the Bayesian Network that underpins inference.

6.3.3. Perform Experiments

For each of the prototypes a number of specific objectives were defined and these were validated experimentally. The results of these evaluations are presented and discussed in detail in Chapter 5. This chapter also posits these results in terms of some selected related work, and identifies a number of strengths and weaknesses in the design. Some of these weaknesses are addressed in the Further Work section below.

The results allow the following conclusions to be drawn.

Intent Inference Engine

The Intent Inference Engine evaluation demonstrates that a user-defined Bayesian Network can be used to aggregate input evidence and arrive at utility-driven inferences on a user's intentions. The engine assumes that the initial pre-processing of sensor data is handled elsewhere, as is the case in similar systems such as S-SEER [Oliver, 2005].

The evaluation notes that for more complex implementations, there would be a need to focus more on the treatment of evidence, and in particular to propose methods for handling ambiguity and the degradation of certain types of evidence over time.

The user feedback also indicates a need to personalise the system on a user-by-user basis to ensure that the system meets the varying expectations of different individuals.

Environment Modelling Tool

This evaluation demonstrates that an ontology-driven dialogue allows users to prepare semantically sufficient and complete models of available resources. The approach highlights the need to have some additional types of system experts who can prepare the required 3-dimensional maps of the target spaces, and also develop the underlying ontology definitions.

The evaluation of the Environment Modelling Tool also demonstrates that by suitably defining the class and attribute names, they can be used to prompt a user to supply additional data that is required by the constraints applied by the ontology. It was also shown that for applications like the modelling tool, a closed worldview is recommended.

Routine Behaviour Description Tool

The evaluation of the RBDT demonstrated that it can produce utility-based Bayesian Network models. It also showed that the level of human effort required, and in particular the amount of explicit and specific information required, was considerably less than that demanded by the JavaBayes tool or by manually specifying the set of values required.

A number of usability issues were identified such as the need to test the models in a third party application, and the inability to decompile networks that have already been produced.

Overall Experimental Conclusions

Each of the three individual components successfully met its design objectives. Overall the evaluation demonstrated that the system provides an intuitive means for supporting the governance requirements and associated usage and management flows of information outlined in Chapter 1. The system also provides an integrated framework where the three components can inter-operate to provide user centric pervasive computing.

In addition, the comparison with related work presented at the end of Chapter 5, and the summary shown in Table 5.4, shows that in terms of its generic approach and breadth of functionality, the system proposed in this thesis offers a number of novel contributions to the state of the art.

6.4. CONTRIBUTION TO THE STATE OF THE ART

6.4.1. Governance of Pervasive Computing Environments

The work presented here offers a new approach to supporting user centric governance in Pervasive Computing Environments. The principle novelty of this contribution lies in its integrated approach to supporting the governance requirements of the various stakeholders in Pervasive Computing Systems. This approach is generic and non-application specific, which contrasts with the application-specific nature of many of the related projects that have been examined.

The approach is also strongly focussed on the needs, expectations and abilities of users as they take on each of the identified governance roles. This focus has resulted in a highly user-centric approach, which seeks to reduce cognitive load. Some of the underlying approaches to providing this intuitiveness are novel in the Pervasive Computing domain, and the demonstration of their worth is another valuable contribution.

6.4.2. Minor Contributions

Autonomic Computing

The work discussed here was presented at the IEEE Autonomic Computing & Communications Workshop 2005 [O'Donnell, 2005]. By addressing challenges in pervasive computing, the work presented here also offers insights for research in Autonomic Computing. In particular, the approach to modelling, and potentially configuring, highly distributed and heterogeneous environments could prove very valuable in Autonomic Computing. Similarly the method used for defining types of activity could also prove transferable.

Ontologies

A number of ontologies have been defined for the evaluation of the prototypes. These could go on to be used by other projects where models of devices are

required. However, the most novel feature of the work presented here is the demonstration of how ontologies can be used to allow the system to work with a user to produce semantically complete models. In the case of the environment configuration tool, the system works with the user to produce a sufficiently complete model. It does this by examining the constraints expressed by the ontology and with the use of a reasoner, it can then prompt the user with meaningful feedback messages such as "Constraint Violation: In 'Person' the property Date_of_Birth must have a value". This approach proved successful in the environment configuration tool.

Capturing Utility-based Models of Behaviour

This thesis contends that capturing models of routine behaviour can help guide the inference of user intent. In order to do this effectively, these models need to be captured such that they contain sufficient information to be useful while also placing a minimum cognitive load on the user. The approach here defines these models in terms of causally related actions where the relationships between nodes reflect a measure of utility. The method by which such models are created simplifies the process of putting knowledge in the model by firstly adopting a graphical approach and then allowing the user to address utility by way of setting thresholds on each arc and indicating the relative influence by way of relative distance. The system then executes an algorithm, which allows for all of the potential output states at a given node to be computed. This means that the user can express their knowledge about routine behaviour in terms of accessible albeit abstracted concepts.

This approach could be applied to more general problems of capturing models of behaviour.

6.5. PUBLICATIONS

This work has contributed to the state of the art through peer-review publications and conference proceedings. These have included:

Title: "Semantics for the Intuitive Governance of Autonomic User-Centred Pervasive Computing Environments", O'Donnell T., Lewis D., Wade V., International Journal of Pervasive Computing and Communications, Vol. 2, Issue 3, Paper 7.

Contribution: This paper presents a summary of the various design decisions, prototype components and evaluation results produced in this thesis.

Title: "Intuitive Human Governance of Autonomic Pervasive Computing", O'Donnell T., Lewis D., Wade V., Proceedings of WoWMoM 2005, pp. 532 - 536, the Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13-16, 2005

Contribution: Presented the results of the Environment Configuration Tool and the Inference Tool as an approach to the governance of pervasive computing environments.

Title: "A Testbed for Evaluating Human Interaction with Ubiquitous Computing Environments", O'Neill E., Klepal M., Lewis D., O'Donnell T., O'Sullivan D., Pesch D., Proceedings of TRIDENCOM'05, Trento (Italy), February 23 - 25, 2005

Contribution: The Environment Configuration Tool was shown as an example of a testbed for evaluating human interaction with pervasive computing environments.

Title: "Managing User-centric Adaptive Services for Pervasive Computing", Lewis, D., O'Donnell, T., Feeney, K., Brady, A., Wade V., Proceeding of the International Conference on Autonomic Computing (ICAC'04), pp. 248-255

Contribution: Presented an intent inference based approach to personalised adaptive service support within the context of a wider constraint-based approach to supporting users in pervasive computing environments.

Title: "Towards an Intuitive Interface for Tailored Service Compositions", Higel S., O'Donnell T., Lewis D., Wade V., Proceedings of DAIS 2003, Lecture Notes in Computer Science 2893, pp. 266-273, International Federation for Information Processing, 2003

Contribution: Presented the intent inference experiment and demonstrated how it might integrate with an adaptive service composition engine.

Title: "Towards a Natural Interface to Adaptive Service Composition", Higel S., O'Donnell T., Wade V., - Proceedings of the 1st International Symposium on Information and Communication Technologies, ACM International Conference Proceeding Series pp. 169-174, 2003

Contribution: Presented the intent inference experiment.

6.6. FUTURE WORK

6.6.1. Expanded environment

The application domain being considered here is pervasive computing; however the tools presented have a more limited scope, operating as they do in simulated environments and subject to many assumptions. A definite avenue for future work would be an expanded test environment.

This expansion could be in 'depth' insofar as the scenarios already examined could be developed into more complete and demanding exemplars, or it could be in 'breadth' by expanding the number of users or evidence handling demands.

Another expansion would be a real world test environment to replace the simulated environments used thus far.

6.6.2. Integrating with pervasive computing middleware

The proposed system assumes that there is stream of suitable input data from sensors and context stores. Addressing this 'suitability' issue would be another potential area for future work. This would require working from a sensor level upwards. At present some work is active in the area of middleware for pervasive computing environments [Trumler, 2005] [Apel, 2004], and an examination of how such middleware could integrate into the work discussed here would be worthwhile.

6.6.3. Service Composition

The system assumes the existence of a service composition platform that can act on the inferences made by the system. At present the system is not integrated into such a platform. Identifying such a platform would be valuable, as it would allow for actual responses in the form of support to be returned to a user based on inferences. In identifying a system, it would also be necessary to agree the format of the requests that are to be dispatched as well as providing some method for errors or requests for further information to be returned from the composition platform. The message format could consider the Belief-Desire-Intention model as a basis for messaging, or some of the existing languages for making web service requests could be examined.

6.6.4. User Modelling

The approach to defining user behaviour and responding to it presented here is very much from an engineering\process standpoint. It would be advantageous for this work to be expanded through the involvement of behavioural scientists. Indeed as work in pervasive computing focuses on usability it will become increasingly necessary to involve experts from beyond computer science to help shape and guide future research.

APPENDIX I. REFERENCES

[90/270/EEC] - Fifth Individual Directive within the meaning of Article 16 (1) of Directive (89/391/EEC), Official Journal n° L 156 of 21.06.1990, p. 14.

[Abowd, 2000] - Abowd G., Mynatt E., "*Charting past, present, and future research in ubiquitous computing*", ACM Transactions on Computer-Human Interaction (TOCHI), v.7 n.1, p.29 - 58, March 2000.

[Alpert, 2003] Alpert, S. R., Karat, J., Karat, C., Brodie, C., Vergo, J. G., "*User Attitudes Regarding a User-Adaptive eCommerce Web Site*", ACM Journal of User Modelling and User-Adapted Interaction Vol. 13, Issue 4 (Nov. 2003), pp. 373 - 396.

[Apel, 2004] Apel S., Böhm K., "*Using Mixins to Build a Flexible Lightweight Middleware for Ubiquitous Computing*", Proceedings of the 4th IEEE Automated Software Engineering Workshop on Software Engineering and Middleware (SEM'04), Linz, Austria, September 2004.

[Azari, 2004] Azari D., Horvitz E., Dumais S., Brill E., "*Actions, Answers, and Uncertainty: A Decision Making Perspective on Web-Based Question Answering*", Information Processing and Management, 40(5), 2004, pp. 849 – 868.

[Baader, 2005] Baader F., Horrocks I., Sattler U., "*Description logics as ontology languages for the semantic web*", Mechanizing Mathematical Reasoning: Essays in Honor of Jörg Siekmann on the Occasion of His 60th Birthday, Lecture Notes in Artificial Intelligence 2605, pages 228 – 248, Springer-Verlag , 2005.

[Baccarelli, 2005] Baccarelli E., Biagi M., Pelizzoni C., Cordeschi N., Garzia F., "*A Power Allocation Algorithm for Throughput Maximization in Mobile Networks*". Proceedings of WoWMoM 2005, the Sixth IEEE International

Symposium on a World of Wireless Mobile and Multimedia Networks, pp. 437 – 445.

[Bagga, 2002] Bagga J., Heinz A., "*JGraph - A Java Based System for Drawing Graphs and Running Graph Algorithms*", Lecture Notes in Computer Science, Volume 2265, Jan 2002, p. 459.

[Banerjee, 2005] Banerjee N., Basu K., Das S. K., "*Adaptive Resource Management for Multimedia Applications in Wireless Networks*", Proceedings of WoWMoM 2005, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, pp. 250 – 257.

[Bardram, 2004] Bardram J., Christensen H., "*Open Issues in Activity-Based and Task-Level Computing*", Proceedings of the First International Workshop on Computer Support for Human Tasks and Activities, volume PB-2004-60 of Centre for Pervasive Computing Technical Reports. University of Aarhus, Vienna, Austria, April 2004.

[Berners-Lee, 2001] Berners-Lee T., Hendler J., Lassila O., "*The Semantic Web*", Scientific American, May 2001, pp. 28 – 37.

[Beveridge, 2003] "*Combining Task Descriptions and Ontological Knowledge for Adaptive Dialogue*", Lecture Notes in Computer Science, Volume 2807, pp. 341 – 348, Springer-Verlag 2003.

[Bihler, 2005] Bihler P., Brunie L., and Scuturici V.-M., "*Modeling User Intention in Pervasive Service Environments*", Proceedings of The 2005 IFIP International Conference on Embedded and Ubiquitous Computing (Nagasaki, Japan), 6-9 Dec. 2005, pp. 977 – 986.

[Blythe, 1999] Blythe J., "*Decision-Theoretic Planning*", AI Magazine, Vol. 20, No. 2, pp. 37 - 54, 1999.

[Bohnenberger, 2005] Bohnenberger T., Jacobs O., Jameson A., and Aslan I. – "*Decision-Theoretic Planning Meets User Requirements: Enhancements and*

Studies of an Intelligent Shopping Guide", Pervasive Computing 2005, LNCS 3468, pp. 279 – 296.

[Boutilier, 2003] Boutilier C., Das R., Kephart J., Tesauro G., Walsh W., "*Cooperative Negotiation in Autonomic Systems using Incremental Utility Elicitation*", Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI 2003), pp. 89 – 97.

[Bradbury, 2003] Bradbury J., Shell J., Knowles C., "*Hands On Cooking: Towards an Attentive Kitchen*", Conference on Human Factors in Computing Systems Ft. Lauderdale, Florida, USA, pp. 996 – 997.

[Brickley 2003] - D. Brickley and L. Miller. FOAF vocabulary specification. In *RDFWeb Namespace Document*. RDFWeb, `xmlns.com`, 2003.

[Brusilovsky, 1998] Brusilovsky P., "*Adaptive Educational Systems on the World Wide Web: A Review of Available Technologies*", Proceedings of the Workshop on WWW-based Tutoring, 4th International Conference on Intelligent Tutoring Systems ITS '98, August 1998.

[Chalmers, 2001] Chamlers, M., Chitson P., "*Bead: Explorations in Information Visualization*", Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 353 – 354.

[Chen, 2004] - Chen H., Finin T., Joshi A., "*An ontology for contextaware pervasive computing environments*", Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review, 2003.

[Chen, 2003] Chen H., Finin T., Joshi A., "An ontology for context-aware pervasive computing environments", The Knowledge Engineering, Volume 18, Issue 3, pp. 197 – 207, ACM September 2003.

[Christopoulou, 2005] Christopoulou E., Goumopoulos C. Kameas A. "*An ontology-based context management and reasoning process for UbiComp*

applications," Proceedings of Joint sOc-EUSAI conference 2005, Grenoble, France, October 2005, pp. 265 – 270.

[Cohen, 2004] Cohen I., Cozman F., Sebe N., Cirelo M., Huang T., "*Semi-supervised Learning of Classifiers: Theory, Algorithms for Bayesian Network Classifiers and Application to Human-Computer Interaction*", IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Dec 2004.

[Collins, 2000] Collins J., Bilot C., Gini M., Mobasher B., "*Mixed-Initiative Decision Support in Agent-Based Automated Contracting*", Fourth International Conference on Autonomous Agents, Barcelona, Spain, June 2000.

[Connelly, 2003] Connelly K., Khalil A., "*Towards Automatic Device Configuration in Smart Environments*", in the Proceedings of UbiSys Workshop 2003, Fifth Annual Conference on Ubiquitous Computing, Seattle, USA, October 12 - 15, 2003.

[Cozman, 2000] Cozman F., "*Generalizing Variable Elimination in Bayesian Networks*", Workshop on Probabilistic Reasoning in Artificial Intelligence, Atibaia, Brazil, November 20, 2000.

[Crosby, 2003] Crosby, M.E. Iding, M.K. Chin, D.N. , "*Research on task complexity as a foundation for augmented cognition*", System Sciences, 2003, Proceedings of the 36th Annual Hawaii International Conference on Systems Sciences, pp. 132 – 140, IEEE Computer Society, 2003.

[Czarkowski, 2003] Czarkowski, M., Kay J., "*How to give the user a sense of control over the personalization of AH?*" in AH2003: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems; Workshop proceedings from the Ninth International Conference on User Modeling.

[Dagger, 2004] Dagger D., Wade V., Conlan O., "*Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT)*", published in the workshop proceedings of the AH'04 conference, AH2004.

[Das, 2004] Das B., "*Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem*", Technical Report for the Australian Department of Defence, http://arxiv.org/ftp/cs/papers/0411/0411034.pdf.

[Dean, 2004] Dean M., Schreiber G. - *OWL Web Ontology Language Reference*, , Editors, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-owl-ref-20040210/.

[DeBra, 1999] DeBra P., Wu H., Houben G.-J., "*User Modelling in Adaptive Hypermedia Applications*", Proceedings of the Interdisciplinaire Conferentie Informatiewetenschap, pp. 10 – 12, Amsterdam 1999.

[Denaro, 2005] Denaro G., Mariani L., Pezzè M., Tosi D., "*Adaptive Runtime Verification for Autonomic Communication Infrastructures*", Proceedings of WoWMoM 2005, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13 - 16, 2005.

[Denaux, 2005(1)] Denaux, R., Aroyo, L., and Dimitrova, V., "*An approach for ontology-based elicitation of user models to enable personalization on the semantic web*", In *Special interest Tracks and Posters of the 14th international Conference on World Wide Web* (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM Press, New York, NY, pp. 1170 – 1171.

[Denaux, 2005(2)] Denaux, R., Aroyo, L., and Dimitrova, V., "*OWL-OLM: Interactive Ontology-based Elicitation of User Models*", Proceedings of the Workshop on Personalisation on the Semantic Web PerSWeb05 held in conjunction with the 10th International Conference on User Modeling July 24 - 30, 2005, Edinburgh, UK, pp. 34 – 46.

[Felder, 1988] Felder R., "*Learning and Teaching Styles in Engineering Education*," Engineering Education 78(7), pp. 674 – 681.

[Friedman-Hill, 2003] Friedman-Hill E., "*Jess in Action: Rule-based Systems in Java*", ISBN 1930110898.

[Fujinami, 2005] Fujinami K., Kawsar F. and Nakajima T. – "*AwareMirror: A Personalized Display Using a Mirror*", Pervasive Computing 2005, LNCS 3468, pp. 315 – 332.

[Gajos, 2002] Gajos, K., Fox H., Shrobe H., "*End User Empowerment in Human-Centered Pervasive Computing*", Proceedings of the International Conference on Pervasive Computing, Pervasive 2002, Lecture Notes in Computer Science, Vol. 2414 pp. 134 – 140, Springer-Verlag 2002.

[Galata, 2001] Galata A., Johnson N., Hogg D., "*Learning Variable-Length Markov Models of Behavior*", International Journal of Computer, Vol. 81, pp. 398 – 413, Springer-Verlag 2001.

[Garlan, 2000] Garlan D., Monroe R., Wile D., "*Acme: Architectural Description of Component-Based Systems*" *Foundations of Component-Based Systems*, Gary T. Leavens and Murali Sitaraman (eds), Cambridge University Press, 2000, pp. 47 – 68.

[Garofolakis, 2006] Garofalakis M., Brown K., Franklin M., Hellerstein J., Wang D., Michelakis E., Tancau L., Wu E., Jeffery S., Aipperspach R., "*Probabilistic Data Management for Pervasive Computing: The Data Furnace Project*", IEEE Data Engineering Bulletin, Vol. 29, No. 1, March 2006 (Special Issue on Probabilistic Data Management), pp. 57 – 63.

[Gatani, 2005] Gatani L., Lo Re G., Gaglio S., "*An Adaptive Routing Protocol for Ad-Hoc Peer-to-Peer Networks*", *Proceedings of WoWMoM 2005*, pp. 532 - 536, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13 - 16, 2005.

[Gelenbe, 2005] Gelenbe E., Gellman M., Loukas G., "*An Autonomic Approach to Denial of Service Defence*", Autonomic Pervasive Computing,

Proceedings of WoWMoM 2005, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13 - 16, 2005.

[Georgeff, 1999] Georgeff M., Pell B., Pollack M., Tambe M., Wooldridge M., "*The Belief-Desire-Intention Model of Agency*", Intelligent Agents V Springer-Verlag Lecture Notes in AI Volume 1365, March 1999.

[Glinton, 2004] Glinton R., Owens S., Giampapa J., Sycara K., Grindle C., Lewis M., "*Terrain-Based Information Fusion and Inference*", Proceedings of the 7th International Conference on Information Fusion, pp. 338 – 345.

[Greenberg, 2005] Greenberg S., "*Collaborative Physical User Interfaces*", **C**ommunication and Collaboration Support Systems, pp. 24 – 42, IOS Press (Amsterdam, The Netherlands), ISBN: 1-58603-514-2, June 2005.

[Grimm, 2000] Grimm R., Anderson T., Bershad B., Wetherall D., "*A System Architecture for Pervasive Computing*", Proceedings of the Ninth ACM SIGOPS European Workshop, pp. 177 – 182, September 2000.

[Grønmo, 2002] Grønmo G., "*Automagic Topic Maps*", Proceedings of XML Europe 2002.

[Groppe, 2005] Groppe J., Mueller W. "*Profile Management Technology for Smart Customizations in Private Home Applications*", Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05).

[Gruber, 1993] "*A translation approach to portable ontology specifications*," Gruber T., Knowledge Acquisition, 5(2), pp. 199 – 220, 1993.

[Haarslev, 2003] Haarslev V., Möller R. "*Racer: A Core Inference Engine for the Semantic Web*", Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), located at the 2nd

International Semantic Web Conference ISWC 2003, Sanibel Island, Florida, USA, October 20, pp. 27 – 36, 2003.

[Higel, 2003] Higel S., O'Donnell T., Lewis D., Wade V., "Towards an Intuitive Interface for Tailored Service Compositions", Proceedings of Distributed Applications and Interoperable Systems, 4th IFIP WG6.1 International Conference, DAIS 2003, Paris, France, November 17-21, 2003, LNCS 2893, pp. 266 – 273.

[Held, 2002] A. Held, S. Buchholz and A. Schill, "*Modeling of Context Information for Pervasive Computing Applications*", Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics.

[HFRG] Human Factors Research Group, North Mall Enterprise Centre, University College Cork, Cork, Ireland.

[Hong, 2004] Hong, J.I., Landay J.A., "*An Architecture for Privacy-Sensitive Ubiquitous Computing*", In Proceedings of The Second International Conference on Mobile Systems, Applications, and Services (Mobisys 2004). Boston, MA, pp. 177 - 189, 2004.

[Horvitz, 2005] Horvitz E., Apacible J., Sarin R., and Liao L., "*Prediction, Expectation, and Surprise: Methods, Designs, and Study of a Deployed Traffic Forecasting Service*", Twenty-First Conference on Uncertainty in Artificial Intelligence, *UAI-2005*, Edinburgh, Scotland, July 2005.

[Horvitz, 2003] Horvitz E., Kadie C., Paek T., Hovel D., "*Models of Attention in Computing and Communication: From Principles to Applications*", Communications of the ACM, March 2003/Vol. 46, No. 3, pp. 52 – 59.

[Horvitz, 1999] Horvitz, E., "*Principles of Mixed-Initiative User Interfaces*", Proceedings of CHI '99, ACM SIGCHI Conference on Human Factors in Computing Systems, Pittsburgh, PA, May 1999, ACM Press. pp 159 – 166.

[Horvitz, 1998] Horvitz E., Breese J., Heckerman D., Hovel D., Rommelse K., "*The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users*", In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, pp. 256 – 265, Madison, WI, 1998.

[Humble, 2003] Humble J., Crabtree A., Hemmings T., Åkesson K.-P., Koleva B., Rodden T., Hansson P., "*Playing with the Bits" User-Configuration of Ubiquitous Domestic Environments*", Proceedings of the Fifth International Conference on Ubiquitous Computing UbiComp 2003, pp. 256 – 263, Springer-Verlang 2003.

[Iqbal, 2005] Iqbal M. M., Gondal I., Dooley L., "*Autonomic Load-Adaptive Optimization of Beacon Exchange Rate for Ubiquitous MANETs*", Autonomic Pervasive Computing, Proceedings of WoWMoM 2005, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13 – 16, 2005.

[Jakobson, 2004] Jakobson G., Buford J. Lewis L. "*Towards an Architecture for Reasoning about Complex Event-Based Dynamic Situations*", International Workshop on Distributed Event-Based Systems (DEBS '04), Proceedings of the 26th International Conference on Software Engineering (ICSE 2004).

[Jeffery, 2006] Jeffery S., Alonso G., Franklin M., Hong W., Widom J., "*Declarative Support for Sensor Data Cleaning*", Proceedings of Pervasive 2006 - The 4th International Conference on Pervasive Computing, Dublin, Ireland, 2006.

[Jiang, 2006] Jiang H., Li X., Liu C., "*Large margin hidden Markov models for speech recognition*", IEEE Transactions on Audio, Speech and Language Processing Vol. 14 Issue 5, IEEE September 2006.

[Jordan, 2002] Jordan P., Siler S., "*Student Initiative and Questioning Strategies in Computer-Mediated Human Tutoring Dialogues*", In Proceedings of ITS 2002 Workshop on Empirical Methods for Tutorial Dialogue Systems, 2002.

[Kagal, 2005] Kagal L., Berners-Lee T., "*Rein: Where Policies Meet Rules in the Semantic Web*", Technical Report, Massachusetts Institute of Technology, September 2005.

[Kagal, 2003] - Kagal L., Finin T., Joshi A., "A Policy Based Approach to Security for the Semantic Web", In 2nd International Semantic Web Conference (ISWC2003), September 2003.

[Kambhampati, 2005] Kambhampati S., "*Source- and User- Adaptive Information Integration*", Proceedings of the Dagstuhl Seminar on Machine Learning for the Semantic Web, 13-18 February 2005.

[Kephart, 2003] Kephart J. O., Chess D. M. "*The vision of autonomic computing*", IEEE Computer Magazine, Janurary 2003.

[Khungar, 2005] Khungar S., Riekki J., "*A context based storage system for mobile computing applications*", ACM Journal of Mobile Computing and Communications Review, Volume 9, Number 1, pp. 64 – 68.

[Knublauch, 2004] Knublauch H., Ray W. Fergerson, Natalya F. Noy, Mark A. Musen, "*The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications*", Third International Semantic Web Conference - ISWC 2004, Hiroshima, Japan (2004).

[Kokkinos, 2005] Kokkinos P., Papageorgiou C., Varvarigos E., "*Energy-Aware Routing in Wireless Ad-Hoc Networks*", Proceedings of WoWMoM 2005, pp. 532 - 536, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13-16, 2005.

[Leclercq, 2004] Leclercq M., Quéma V., Stefani J.-B., "*DREAM: a component framework for the construction of resource-aware, reconfigurable MOMs*", ACM International Conference Proceeding Series, Vol. 80, Proceedings of the 3rd workshop on Adaptive and reflective, pp. 250 – 255.

[Lenat, 1990] - Lenat D. B., Guha R. V., "*Building Large Knowledge-Based Systems: Representation and Inference in the CycProject*", Addison-Wesley, February 1990.

[Lewis, 2004] Lewis, D., O'Donnell, T., Feeney, K., Brady, A., Wade V., "*Managing User-centric Adaptive Services for Pervasive* Computing", Proceedings of the International Conference on Autonomic Computing, New York, 2004, IEEE.

[Lewis, 2004(2)] Lewis D., Feeney K., Carey K., Tiropanis T., Courtenage S., "*Semantic-based Policy Engineering for Autonomic Systems*", in Proc of 1st IFIP WG6.6 International Workshop on Autonomic Communication - Autonomic Communication Principles, Berlin 18-19 October 2004.

[Luckham, 2005] Luckham D. "*Achieving Instant Insight into the Real-time Electronic Enterprise*", Technical Paper, Dept. of Computer Science, Stanford University.

[Luckham, 2003] Luckham D., Manens A., Bhansali S, Park W., Daswani S., "*Modeling and Causal Event Simulation of Electronic Business Processes*", Proceeding of The 6th Annual Hawaii International Conference on Business.

[Luckham, 2002] Luckham, D., "*The Power of Events: An Introduction to Complex Event Processing*", Pearson Education Inc, 2002.

[Luckham, 1996] Luckham D. "*Rapide: A Language and Toolset for Simulation of Distributed Systems by Partial Orderings of Events*", DIMACS Partial Order Methods Workshop IV, Princeton University, July 1996.

[McGinty, 2003] McGinty L., Smyth B., "*Intention Recognition for Mixed-initiative Recommender Systems*", Proceedings of Eighteenth International Joint Conference on Artificial Intelligence, pp. 94 – 99.

[McGuinness, 2002] McGuinness D., "*Description Logic for Configuration*". Chapter 12 of The Description Logic Handbook: Theory, Implementation, and Applications (Eds. Baader F., McGuinness D., Nardi D., Patel-Schneider P.),. Cambridge University Press, 2002

[Meier, 2003] Meier W., "eXist*: An Open Source Native XML Database*", Proceedings of Web, Web-Services, and Database Systems: NODe 2002, Web- and Database-Related Workshops, Lecture Notes in Computer Science Vol. 2593, Springer-Verlag 2003.

[Mises, 1966] Mises L., "*Human Action: A Treatise on Economics*", Henry Regnery Company 1966.

[Moënne-Loccoz, 2003] Moënne-Loccoz N., Brémond F., Thonnat M., "*Recurrent Bayesian Network for the Recognition of Human Behaviors from Video*", Proceedings of the Third International Conference On Computer Vision Systems (ICVS 2003), Graz, Austria, April 2003, April 1-3, 2003., pp. 68 – 77.

[Morris, 2005] Morris M., Intille S., Beaudin J., "*Embedded Assessment: Overcoming Barriers to Early Detection with Pervasive Computing*", Pervasive Computing 2005, LNCS 3468, pp. 333 – 346.

[Mühlhaüser, 2000] Mühlhaüser M., "*Ubiquitous Computing and Its Influence on MSE*", Proceedings of International Symposium on Multimedia Software Engineering 2000, pp. 48 – 58.

[Neustaedter, 2005] Neustaedter C., Elliot K., Greenberg S. "*Understanding Interpersonal Awareness in the Home*", ACM CHI 2005 Workshop on Awareness systems: Known Results, Theory, Concepts and Future Challenges.

[Nixon, 2002] Nixon P., Dobson S., Terzis S., Wang F., "*Architectural Implications for Context Adaptive Smart Spaces*", IEEE International Workshop on Networked Appliances, 2002.

[Noy, 2001] Noy N., Sintek M., Decker S., Crubezy M., Fergerson R., Musen M., "*Creating Semantic Web Contents with Protege-2000*", IEEE Intelligent Systems 48(2), pp. 60 – 71, 2001.

[O'Donnell, 2005] O'Donnell T., Lewis D., Wade V., "*Intuitive Human Governance of Autonomic Pervasive Computing*", Proceedings of WoWMoM 2005, pp. 532 – 536, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13 - 16, 2005.

[Oliver, 2005] Oliver N., Horvitz E., "*A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities*", Proceedings of 10th International Conference on User Modelling (UM'2005), Lecture Notes on Artificial Intelligence LNAI 3538, pp. 205 – 215, Springer-Verlag 2005.

[Oliver, 2003] Oliver N., Horvitz E., "*Selective Perception Policies for Guiding Sensing and Computation in Multimodal Systems: A Comparative Analysis*", Fifth International Conference on Multimodal Interfaces, November 2003.

[Pacifici, 2003] Pacifici G., Spreitzer M., Tantawi A., Youssef A., "*Performance Management for Web Services*", Research Report RC22676, IBM T.J. Watson Research Center, 2003.

[Pan, 2004] Pan F., Hobbs J. R., "*Time in OWL-S*", In Proceedings of AAAI-04 Spring Symposium on SemanticWeb Services, Stanford University, California, 2004.

[Parsia, 2004] Parsia B., Sirin E., "*Pellet: An owl dl reasoner*", Proceedings of the International Workshop on Description Logics (DL2004), June 2004.

[Perich, 2005] Perich F., Joshi A., Yesha Y., Finin T. "*Collaborative joins in a pervasive computing environment*", The VLDB Journal — The International Journal on Very Large Data Bases, Volume 14 , Issue 2 (April 2005), pp. 182 – 196.

[Perich, 2004] - F. Perich. MoGATU BDI Ontology, 2004.

[Poltrock, 1986] Poltrock S. E. , Steiner D. D. , Tarlton P. N., "*Graphic interfaces for knowledge-based system development*", Conference on Human Factors in Computing Systems, Proceedings of the SIGCHI conference on Human factors in computing systems, Boston, Massachusetts, United States, pp. 9 – 15.

[Rabiner, 1993] Rabiner L., Huang B., "*Fundamentals of Speech Recognition*", Prentice-Hall, 1993.

[Rabiner, 1989] Rabiner L., "*A tutorial on hidden Markov models and selected applications in speech recognition*", Proceedings of the IEEE Vol. 77, pp. 257–286, IEEE 1989.

[Randell, 1992] - Randell D. A., Cui Z., Cohn A., "*A spatial logic based on regions and connection*". In B. Nebel, C. Rich, and W. Swartout, editors, *KR'92*. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, pp. 165–176. Morgan Kaufmann, San Mateo, California, 1992.

[RDF] Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.

[Rimmer, 2005] Rimmer J., Owen T., Wakeman I., Keller B., Weeds J., Wir D., "*User Policies in Pervasive Computing Environments*", Workshop on User Experience Design for Pervasive Computing, Proceedings of Pervasive Computing 2005, May 2005, Munich, Germany.

[Rodden, 2004] Rodden, T. Crabtree, A., Hemmings, T., Koleva, B., Humble, J., Åkesson, K. and Hansson, P., "*Configuring the Ubiquitous Home*", Cooperative Systems Design: Scenario-Based Design of Collaborative Systems

(eds. Darses, F., Dieng, R., Simone, C. and Zacklad, M.), pp. 227 – 241, Amsterdam: IOS Press.

[Rossi, 2002] – Rossi P., Tavoni M., Cocco F., Marschinski R., "*Portfolio Selection with Probabilistic Utility, Bayesian Statistics, and Markov Chain Monte Carlo*," Finance 0211003, Economics Working Paper Archive at WUSTL.

[Rosti, 2002] Rosti A.-V.I., Gales M., "*Factor analysed hidden Markov models*", Proceedings of Acoustics, Speech, and Signal Processing, 2002. Proceedings. (ICASSP '02), Vol. 1, pp. 949 – 952, IEEE 2002.

[Sankoh, 2002] - Sankoh O. A., Becher H., " *Bayesian analysis of selected risk factors of childhood mortality in rural Burkina Faso using a logistic disposition model*", *Department of Tropical Hygiene and Public Health, University of Heidelberg Im Neuenheimer Feld 324, D-69120 Heidelberg, Germany.*

[Satoh, 2003] Satoh I., "*Location-Based Services in Ubiquitous Computing Environments*", Lecture Notes in Computer Science, Volume 2910, Jan 2003, pp. 527 – 542.

[Schuster, 1999] Schuster A., Sterrit R., Admson K., Shapcott M., Curran E. "*Case-based Reasoning for Complex Telecommunication Systems*" Proceedings of 1st European Symposium on the Application of Soft Computing in Telecommunication.

[Shell, 2003] Shell J., Selker T., Vertegaal R. "*Interacting with Groups of Computers*" Communications of the ACM, Vol. 46, No. 3, pp. 40 – 46.

[Shen, 2004] Shen, C., Vernier, F.D., Forlines, C., Ringel, M., "*DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction",* ACM Conference on Human Factors in Computing Systems (CHI), ISBN: 1-58113-702-8, pp. 167 – 174,, April 2004.

[Sloman, 2001] Sloman M., "Will Pervasive *Computers be Manageable*", Keynote Speech, HP OpenView 2001, New Orleans, June 2001.

[Sousa, 2005] Sousa J. P., Poladian V., Garlan D., Schmerl B., Shaw M., "*Task-based Adaptation for Ubiquitous Computing*", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Special Issue on Engineering Autonomic Systems.

[Sterelny, 2004] Sterelny, K., "*Cognitive Load and Human Decision, or Three Ways of Rolling the Rock up Hill*", The Innate Mind: Culture and Cognition, Cambridge University Press, 2004.

[Streitz, 2005] Streitz, N., Nixon, P, "*Introduction to Special Edition on the Disappearing Computer*", Communications of the ACM, Vol. 48, No. 3, March 2005, pp. 32 – 35.

[Sweller, 1998] Sweller, J., Van Merrienboer, J., Paas, F., "*Cognitive Architecture and Instructional Design*", Educational Psychology Review, 10 (3), pp. 251 – 296.

[Sweller, 1988] Sweller, J., "*Cognitive load during problem solving: Effects on learning*", Cognitive Science, 12, pp. 257 – 285.

[Taffat- Bouzid, 2005] Tafat-Bouzid A., Courant M., Hirsbrunner B., "*Implicit Environment-based Coordination in Pervasive Computing*", 20th ACM Symposium on Applied Computing, SAC'05, Santa Fe, New Mexico, USA, 13 - 17 March 2005.

[Tan, 2005] – Tan J.G., Zhang D., Wang X., Cheng H. S., "*Enhancing Semantic Spaces with Event-Driven Context Interpretation*", Pervasive Computing 2005, LNCS 3468, pp. 80 – 97.

[Thomson, 2005] Thomson G., Nixon P., Terzis S., "*Situation Determination with Distributed Context Histories*", 1st International Workshop on Exploiting

Context Histories in Smart Environments. 3rd International Conference on Pervasive Computing - PERVASIVE 2005. Munich, Germany. May 2005.

[Trumler, 2005] Trumler W., Bagci F., Petzold J., Ungerer T., "*AMUN - autonomic middleware for ubiquitous environments applied to the smart doorplate*", ELSEVIER Advanced Engineering Informatics, Volume 19 Issue 3, pp. 243 – 252, 2005.

[Uschold, 1999] Uschold M., Jasper R., "*A Framework for Understanding and Classifying Ontology Applications*", Proceedings of the Twelfth Workshop on Knowledge Acquisition Modeling and Management KAW 99

[Vertegall, 2003] Vertegall, R., "*Attentive User Interfaces*", Communications of the ACM, March 2003/Vol. 46, No. 3, pp. 30 – 33.

[Wang, 2005] Wang H., Horridge M., Rector A., Drummond N., Seidenberg J., "*Debugging OWL-DL Ontologies: A Heuristic Approach*", Proceedings of the 4th International Semantic Web Conference, November 2005.

[Wasinger, 2005] Wasinger R., Krüger A. Jacobs O., "*Integrating Intra and Extra Gestures into a Mobile and Multimodal Shopping Assistant*", Pervasive Computing, 2005, LNCS 3468, pp. 297– 314.

[Weeds, 2004] Weeds J., Keller B., Weir D., Wakeman I., Rimmer J., Owen T., "*Natural Language Expression of User Policies in Pervasive Computing Environments*", Proceedings of OntoLex 2004 (LREC Workshop on Ontologies and Lexical Resources in Distributed Environments). Lisbon, Portugal. May 2004.

[Weiser, 1993] Weiser M., "*Some computer science issues in ubiquitous computing*", Communications of the ACM, v.36 n.7, pp. 75 – 84, July 1993.

[Weiser, 1991] Weiser, M., "*The Computer of the 21st Century*", Scientific American, vol. 265, no.3, September 1991, pp 66 – 75.

[Witbrock, 2003] Witbrock M., Baxter D., Curtis J., Schneider D., Kahlery R., Miraglia P., Wagner P., Panton K., Matthews G., Vizedom A. "*An Interactive Dialogue System for Knowledge Acquisition in Cyc*", Proceedings of Eighteenth International Joint Conference on Artificial Intelligence, pp. 138 – 145.

[Wu, 2002] Wu H., Siegel M., Ablay S., "*Sensor Fusion for Context Understanding*", the Proceedings of IMTC'2002, (Anchorage, AK, USA, 21-23 May 2002).

[Yau, 2004] Yau S., "*Embedded Software in Real-time Pervasive Computing Environments*", Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC'04), pp 406 – 407.

[Yousef, 2005] Yousef B., Hoang D., Rogers G., "*Autonomic Resource Management for Extensible Control Plane*", Proceedings of WoWMoM 2005, pp. 532 - 536, The Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, Taormina\Giardini Naxos (Italy), June 13 - 16, 2005.

[Zeidler, 2001] Zeidler A., "*User Interface Design for Ubiquitous Computing - W@PNotes, an Example*", Workshop on Distributed and Disappearing UIs in Ubiquitous Computing, Proceedings of CHI2001.

APPENDIX II. INTENT INFERENCE ENGINE

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   YES NO

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   VERY HAPPY HAPPY NEUTRAL UNSATISFIED VERY UNSATISFIED

   _____

   _____

   _____

3. What changes would you make in order to improve the support offered?

   _____

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

_____

_____

_____

General Comments

_____

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   ~~YES~~ NO

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   ~~VERY HAPPY~~ HAPPY NEUTRAL UNSATISFIED VERY UNSATISFIED
   Very happy, the system was right every time

3. What changes would you make in order to improve the support offered?

   _None

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

   ___I would find talking the easiest, it's most like how I would look for help

   _____

   _____

General Comments

_____

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   YES NO 2 yes 1 no

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   VERY HAPPY HAPPY ~~NEUTRAL~~ UNSATISFIED VERY UNSATISFIED
   I shouldn't have to start again when the system gets it wrong.

   _____

   _____

   _____

3. What changes would you make in order to improve the support offered?

   __I'd like more explanation when I don't get the expected support

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

__Speech and hands_____

_____

_____

General Comments

_Good idea but I think you get more out of a system that used real sensors and support

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   ~~YES~~ NO

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   ~~VERY HAPPY~~ HAPPY NEUTRAL UNSATISFIED VERY UNSATISFIED

   Good stuff_____

   _____

   _____

3. What changes would you make in order to improve the support offered?

   _I'd like more ways of giving instructions_____

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?
    - Talking
    - Gesturing
    - Using keyboards or textpads

General Comments

_____

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   ~~YES~~ NO

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   VERY HAPPY ~~HAPPY~~ NEUTRAL UNSATISFIED VERY UNSATISFIED

   __The support was good, but I still like the idea of doing things in a more hands on or tactile way. Maybe I just need to get used to it!

   _____

   _____

3. What changes would you make in order to improve the support offered?

   _More ways of providing inputs? _____

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

   _For the phonecall I'd probably prefer to hold the phone or mic, or even sit beside it. Not sure I'd like to talk into the air where everyone could hear me.
   The others were good.

   _____

   _____

General Comments

_Look forward to seeing a real version _____

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   ~~YES~~ NO

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   ~~VERY HAPPY~~ HAPPY NEUTRAL UNSATISFIED VERY UNSATISFIED

   _Yes – although the responses could have been friendlier, but maybe not the paperclip

   _____

   _____

3. What changes would you make in order to improve the support offered?

   Some work on the messages that come back_____

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

__Gestures and voice_____

_____

_____

General Comments

_____

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   YES NO 2 Yes 1 No

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   VERY HAPPY ~~HAPPY~~ NEUTRAL UNSATISFIED VERY UNSATISFIED

   I think the one it got wrong was because I changed my mind about how to do it in the middle on the file moving
   exercise_____

   _____

   _____

3. What changes would you make in order to improve the support offered?

   _The ability to change my mind!_____

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

   _Voice, hand gestures, tone of voice, position,
   gaze_____

   _____

   _____

General Comments

_____

_____

_____

Questionnaire

You have just attempted to use the office simulator. Based on this interaction, could you please answer the following.

1. Did the system successfully deduce your tasks?
   ~~YES~~ NO

2. How satisfied were you with the support, correct or incorrect? Please comment on your mark.
   ~~VERY HAPPY~~ HAPPY NEUTRAL UNSATISFIED VERY UNSATISFIED

   __Very Happy 100%_____

   _____

   _____

3. What changes would you make in order to improve the support offered?

   _Maybe a 3d office rather than the picture _____

   _____

   _____

And finally.

4. If this test scenario was a real world system, what type of interaction would you find most natural and useful for each of the tasks?

__Probably voice and pointing_____

_____

_____

General Comments

_____

_____

_____

APPENDIX III. ENVIRONMENT CONFIGURATION TOOL

Questionnaire 1 – Pre-conceptions

Name:

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES _____ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES _____ NO _____

3. How long did this take?

_____ minutes

4. Could you briefly describe the steps involved in preparing the room?

Questionnaire 1 – Pre-conceptions (Results)

Name: Participant 1

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES __✓__ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES __✓__ NO _____

3. How long did this take?

_95_ minutes

4. Could you briefly describe the steps involved in preparing the room?

- Install laptop
- Power blocks & plugs
- Wired network provision and wireless
- [Microsoft] Office installation
- Browser config
- File migration

Questionnaire 1 – Pre-conceptions (Results)

Name: Participant 2

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES __✓__ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES __✓__ NO _____

3. How long did this take?

_240_ minutes

4. Could you briefly describe the steps involved in preparing the room?

- Finding appropriate device drivers for computer equipment
- Installing drivers on multiple machines
- Setting up several PC's with appropriate permissions\configuration
- Issuing usernames\passwords to attendees
- Configuring projector for presentation

Questionnaire 1 – Pre-conceptions (Results)

Name: Participant 3

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES __✓__ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES __✓__ NO _____

3. How long did this take?

_10_ minutes

4. Could you briefly describe the steps involved in preparing the room?

- Tested for wireless signal
- Plugged in wireless cards
- Setup projector
- Setup microphone

Questionnaire 1 – Pre-conceptions (Results)

Name: Participant 4

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES __✓__ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES __✓__ NO _____

3. How long did this take?

_10_ minutes

4. Could you briefly describe the steps involved in preparing the room?

- Ensured [Microsoft] Powerpoint display on a laptop (sic.)
- Computer was correctly displayed using a laptop (sic.)

Questionnaire 1 – Pre-conceptions (Results)

Name: Participant 5

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES __✓__ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES __✓__ NO _____

3. How long did this take?

_5__ minutes

4. Could you briefly describe the steps involved in preparing the room?

- Configured the laptop for wireless access
- Insuring one of the laptops had access had access to the projector so the speakers could present the class

Questionnaire 1 – Pre-conceptions (Results)

Name: Participant 6

1. Have you experience of preparing a room for a business function – e.g. presentation, conference call, team meeting?

YES __✓__ NO _____

If so, please answer the following:

2. Did you have to install or configure any particular equipment for this event?

YES __✓__ NO _____

3. How long did this take?

40 to 180 minutes

4. Could you briefly describe the steps involved in preparing the room?

(2 occasions trying to organise laptop access for visiting researchers at TCD)

- On first occasion the prescribed configuration did not work due to human error in the instructions, i.e. the wrong network name was supplied
    - To identify and solve the task took 3 to 4 hours
- 2nd occasion, configuration was delayed by up to 40 mins waiting for a technician to validate MAC address

Questionnaire 2 – SUMI Feedback & Aggregate Results

	AGREE	NEUTRAL	DISAGREE
1. I would recommend this software	6	0	0
2. This software has stopped unexpectedly	5	0	1
3. Learning to use this software initially is full of problems	1	2	3
4. I sometimes don't know what to do next	2	1	3
5. I enjoyed my sessions with this software	5	1	0
6. If this software stops it is not easy to restart it	0	3	3
7. It takes too long to learn the commands	0	0	6
8. I sometimes wonder if I'm using the right commands	2	0	4
9. Working with this software is satisfying	6	0	0
10. The system info is clear and understandable	4	1	1
11. The software interrupts the way I normally do my work	1	3	2
12. Working with this software is mentally stimulating	5	1	0
13. There is never enough info on screen when it's needed	1	0	5
14. I feel in command of this software	4	2	0
15. I think this software is inconsistent	0	1	5
16. I would not like to use this software every day	0	1	5
17. I can understand the information provided by this	6	0	0

software			
18. This software is awkward when I want to do something which is not standard	1	2	3
19. Tasks can be performed in straightforward manner using this soft	5	1	0
20. Using this software is frustrating	6	0	0
21. This software has helped me overcome problems I have had using it	1	4	1
22. The speed of this software is fast enough	4	1	1
23. I keep having to ask questions from the demonstrator	2	2	2
24. It is obvious that user needs have been fully taken into consideration	3	2	1
25. There have been times when using this software has made me tense	1	0	5
26. The organisation of this software seems quite logical	5	1	0
27. The software allows economy of keystrokes	5	1	0
28. Learning to use the system is difficult	0	2	4
29. There are too many steps to get something to work	1	0	5
30. I think this software has given me a headache on occasion	0	1	5
31. It is easy to make this software do exactly what I want	4	1	1
32. I will never learn to use all that is offered in this	6	0	0

software			
33. The software hasn't always done what I expected	2	1	3
34. The software has a very attractive presentation	3	3	0
35. It is relatively easy to move from one part of a task to another	6	0	0
36. It is easy to forget how to do things with this software	0	2	4
37. The software occasionally behaves in a way which can't be understood	2	1	3
38. The software is really awkward	6	0	0
39. It is easy to see at a glance what the options are at each stage	2	4	0
40. Getting data files in and out is not easy	0	1	5

Questionnaire 3 – Post Conceptions

Name:

1. Are there any additional features you would like to see in this tool, and if so, what are they?

2. Have you any other comments?

Questionnaire 3 – Post Conceptions (Results)

Name: Participant 1

1. Are there any additional features you would like to see in this tool, and if so, what are they?

- Confirmation for each device edited
- Should a device be able to have itself as a target?

2. Have you any other comments?

- Rename error messages when they are not programme errors

Questionnaire 3 – Post Conceptions (Results)

Name: Participant 2

1. Are there any additional features you would like to see in this tool, and if so, what are they?

- Hierarchical representation of entity relationships in GUI
- More scope for describing the entity
- Set of 3D entities in Hammer editor (a computer for a computer!) would remove the resizing issues and give more contextual information to the user when viewing the 3D environment

2. Have you any other comments?

Questionnaire 3 – Post Conceptions (Results)

Name: Participant 3

1. Are there any additional features you would like to see in this tool, and if so, what are they?

- Interrelated entities
- Entity specific errors rather than all together
- More integration between pieces of software

2. Have you any other comments?

- Hammer editor is sometimes clunky
- Design tool is sometimes too eager to get rid of entities

Questionnaire 3 – Post Conceptions (Results)

Name: Participant 4

1. Are there any additional features you would like to see in this tool, and if so, what are they?

2. Have you any other comments?

- Map software is fairly intuitive to use
- Will be very easy to when help or documentation is added
- Design tool good for ensuring all bases are covered

Questionnaire 3 – Post Conceptions (Results)

Name: Participant 5

1. Are there any additional features you would like to see in this tool, and if so, what are they?

- Visual representation of the device hierarchy

2. Have you any other comments?

- No

Questionnaire 3 – Post Conceptions (Results)

Name: Participant 6

1. Are there any additional features you would like to see in this tool, and if so, what are they?

- Modification of partial instances

2. Have you any other comments?

APPENDIX IV. RESOURCE DEFINITION FRAMEWORK (RDF)

The Resource Description Framework (RDF) provides a lightweight approach to handling metadata in XML documents. It was developed by the W3C at the same time as XML.

In RDF the object being described is referred to as a *resource*, and each resource has one or more attributes referred to as *properties*. RDF also defines the notion of a *statement*, which is a resource together with a property and its value for that resource. Each statement has a subject, a predicate and an object. For example, an RDF statement representing a pizza with a ham topping might take the form:

Subject	Pizza
Predicate	hasTopping
Object	Ham

Table IV.1 – The Subject-Predicate-Object relationship in RDF

```
<rdfs:Class rdf:about="Pizza"
       rdfs:label="Pizza">
      <rdfs:subClassOf rdf:resource="Resource"/>
</rdfs:Class>
```

Table IV.2 RDF Class Definition

```
<rdf:Property rdf:about= hasTopping"
       a:maxCardinality="1"
       rdfs:label="hasTopping">
      <rdfs:domain rdf:resource="Pizza"/>
      <rdfs:range rdf:resource="Literal"/>
</rdf:Property>
```

Table IV.3 – RDF Property Definition

```
<Pizza rdf:about="HamePizza" kb:hasTopping="Ham"
       rdfs:label="HamPizza"/>
```

Table IV.4 – RDF Instance

The object can either be a literal value, a simple integer say, or it can be another RDF resource referred to by a URI. The object 'Ham' could therefore be represented by a URI pointing to an RDF description of ham or just by the string 'Ham'.

RDF also makes use of namespaces to allow for properties defined elsewhere to be shared amongst ontologies. This would allow a property *hamTopping* defined at *http://pizza.org/hamTopping* to be referred as follows:

```
<rdf xmlns:top= http://pizza.org/hamTopping/>
```

Now when the property *hamTopping* is referred to it can be given the qualified name *top:hamTopping* which will map it to the concept specified via the URI above.

The RDF framework provides the basis for many of the ontology languages currently in use.

DARPA Agent Mark-up Language – DAML

The lightweight nature of RDF means it is an excellent basis for a variety of application domains, however it often needs to be extended when it is being applied. The increasing demands of the Semantic Web meant that such an extension was needed, and the US Defence Advance Research Projects Agency (DARPA) proposed their own DAML mark-up language to fulfil this requirement.

The first product of the DAML project was DAML-ONT, which was first proposed in 2000. This was further developed by its combination with the Ontology Inference Layer to produce DAML+OIL. DAML+OIL is now the main implementation of DAML.

One of the first improvements DAML+OIL makes on simple RDF is to take a more restrictive approach to datatypes. As mentioned above, RDF allows properties to have either literal values or another RDF resource. However, in the case of literals, RDF doesn't make any further attempt to classify the datatype. So a literal value '5' could be a string or an integer. In the semantic web domain, knowing the datatype is an important part of understanding its significance and so DAML+OIL addresses this ambiguity.

DAML+OIL restricts literal values to custom types and those defined in the XML Schema Definition Language (XSDL). Some of the XSDL datatypes are listed in Table IV.5.

These datatypes can be referenced using the namespace feature of RDF.

Data Type	Description
String	The **string** datatype represents character strings in XML.
Boolean	The **Boolean** datatype has the value space required to support the mathematical concept of binary-valued logic: {true, false}.
Number	The **number** datatype is the standard mathematical concept of number, including the integers, reals, rationals, etc.
dateTime	The **dateTime** datatype represents a combination of date and time values as defined in the SQL standard and in ISO 8601 encoded as a single string.
Binary	The **binary** datatype represents strings (blobs) of binary data.
URI	The **uri** datatype represents a standard Universal Resource Identifier reference
Integer	The **integer** datatype corresponds to the standard mathematical concept of integer numbers.
Decimal	The **decimal** datatype restricts allowable values to numbers with an exact fractional part.
Real	The **real** datatype is a computational approximation to the standard mathematical concept of real numbers.
Date	The **date** datatype represents a date value as defined in ISO8601 encoded as a single string.
Time	The **time** datatype represents a time value as defined in [ISO 8601] encoded as a single string.
TimePeriod	The **timePeriod** datatype represents a period of time as defined in ISO 8601 encoded as a single string.

Table IV.5 – XSDL Datatypes

As well as giving more information on the type of the data being described, DAML+OIL also allows for further restrictions on how the ontology is instantiated. For example, DAML+OIL allows for restrictions to be enforced on properties such as that they must be unique or that a given property has a cardinality restriction.

Another important additional feature of DAML+OIL is that it provides an easy method of describing enumerations. This means that with DAML+OIL you can limit the number of potential values for a property, so for a wine ontology with a *hasGrapeVariety* property, the possible values could be limited to *shiraz*, *pinot noir* and *cabernet sauvignon* where these are instances of a grape variety ontology.

Finally, DAML+OIL allows for a number of operations from set theory to be employed to link objects. These include inverses and transitivity. Including these relationships allows for collections of objects to be classified into taxonomies and for links that weren't explicitly set down to be inferred by viewing these relationships. For example, if a model stated that any pizza with meat was a meaty pizza and that ham was a meat, a reasoner could classify a ham pizza into the meaty pizza taxonomy without needing to be told deliberately that the ham topping is a meat.

APPENDIX V. AN OWL ENVIRONMENT MODEL

The OWL model below represents a room containing a number of devices. The instances of the various devices are towards the end, while the opening mark-up defines the ontology itself.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:doc="http://pervasive.semanticweb.org/ont/2004/06/document#"
   xmlns="http://www.owl-ontologies.com/unnamed.owl#"
   xmlns:mtg="http://pervasive.semanticweb.org/ont/2004/06/meeting#"
   xmlns:know="http://pervasive.semanticweb.org/ont/2004/06/knowledge#"
   xmlns:sch="http://pervasive.semanticweb.org/ont/2004/06/schedule#"
   xmlns:ddc="http://pervasive.semanticweb.org/ont/2004/06/digital-doc#"
   xmlns:rcc="http://pervasive.semanticweb.org/ont/2004/06/rcc#"
   xmlns:dev="http://pervasive.semanticweb.org/ont/2004/06/device#"
   xmlns:per="http://pervasive.semanticweb.org/ont/2004/06/person#"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
   xmlns:eve="http://pervasive.semanticweb.org/ont/2004/06/event#"
   xmlns:agent="http://pervasive.semanticweb.org/ont/2004/06/agent#"
   xmlns:pol="http://pervasive.semanticweb.org/ont/2004/06/policy#"
   xmlns:loc="http://pervasive.semanticweb.org/ont/2004/06/location#"
   xmlns:act="http://pervasive.semanticweb.org/ont/action#"
   xmlns:p1="http://pervasive.semanticweb.org/ont/2004/06/action#"
   xmlns:tme1="http://www.isi.edu/~pan/damltime/time-entry.owl#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:icap="http://pervasive.semanticweb.org/ont/2004/06/img-capture#"
   xmlns:bdi="http://pervasive.semanticweb.org/ont/2004/06/bdi#"
   xmlns:geom="http://pervasive.semanticweb.org/ont/2004/06/geo-
measurement#"
   xmlns:foaf="http://xmlns.com/foaf/0.1#"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:spc="http://pervasive.semanticweb.org/ont/2004/06/space#"
```

```
   xmlns:tme="http://pervasive.semanticweb.org/ont/2004/06/time#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about="">
   <owl:imports
rdf:resource="http://pervasive.semanticweb.org/ont/2004/06/location"/>
   <owl:imports
rdf:resource="http://pervasive.semanticweb.org/ont/2004/06/person"/>
  </owl:Ontology>
  <owl:Class rdf:ID="PortableComputingDevice">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="ComputingDevice"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="VoiceCommunicationDevice">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="CommunicationsDevice"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="SensorDevice">
   <rdfs:subClassOf>
    <owl:Class rdf:ID="Device"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="LaptopComputingDevice">
   <rdfs:subClassOf rdf:resource="#PortableComputingDevice"/>
  </owl:Class>
  <owl:Class rdf:ID="DisplayDevice">
   <rdfs:subClassOf>
    <owl:Class rdf:about="#Device"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="PDAComputingDevice">
   <rdfs:subClassOf rdf:resource="#PortableComputingDevice"/>
  </owl:Class>
  <owl:Class rdf:ID="Wired">
```

```
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Protocol"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Wireless">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Protocol"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="ScreenDevice">
    <rdfs:subClassOf rdf:resource="#DisplayDevice"/>
  </owl:Class>
  <owl:Class rdf:ID="ExplicitInputDevice">
    <rdfs:subClassOf rdf:resource="#SensorDevice"/>
  </owl:Class>
  <owl:Class rdf:about="#CommunicationsDevice">
    <rdfs:subClassOf>
      <owl:Class rdf:about="#Device"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="MouseDevice">
    <rdfs:subClassOf rdf:resource="#ExplicitInputDevice"/>
  </owl:Class>
  <owl:Class rdf:about="#Protocol">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:ID="hasStandard"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
```

```
  </owl:Class>
  <owl:Class rdf:ID="Telephone">
   <rdfs:subClassOf rdf:resource="#VoiceCommunicationDevice"/>
   <rdfs:subClassOf>
    <owl:Restriction>
     <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
     >1</owl:minCardinality>
     <owl:onProperty>
      <owl:DatatypeProperty rdf:ID="hasPhoneNumber"/>
     </owl:onProperty>
    </owl:Restriction>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="DesktopComputerDevice">
   <rdfs:subClassOf>
    <owl:Class rdf:about="#ComputingDevice"/>
   </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="#Device">
   <rdfs:subClassOf>
    <owl:Restriction>
     <owl:someValuesFrom
rdf:resource="http://xmlns.com/foaf/0.1#Person"/>
     <owl:onProperty>
      <owl:ObjectProperty rdf:ID="hasAllowedUsers"/>
     </owl:onProperty>
    </owl:Restriction>
   </rdfs:subClassOf>
   <rdfs:subClassOf>
    <owl:Restriction>
     <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
     >1</owl:cardinality>
     <owl:onProperty>
```

```
        <owl:ObjectProperty rdf:ID="hasLocation"/>
       </owl:onProperty>
      </owl:Restriction>
     </rdfs:subClassOf>
     <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
     <rdfs:subClassOf>
      <owl:Restriction>
       <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasAllowedUsers"/>
       </owl:onProperty>
       <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
       >1</owl:minCardinality>
      </owl:Restriction>
     </rdfs:subClassOf>
    </owl:Class>
    <owl:Class rdf:ID="FixedLineTelephone">
     <rdfs:subClassOf rdf:resource="#Telephone"/>
    </owl:Class>
    <owl:Class rdf:ID="KeyboardDevice">
     <rdfs:subClassOf rdf:resource="#ExplicitInputDevice"/>
    </owl:Class>
    <owl:Class rdf:ID="ProjectionDevice">
     <rdfs:subClassOf rdf:resource="#DisplayDevice"/>
    </owl:Class>
    <owl:Class rdf:about="#ComputingDevice">
     <rdfs:subClassOf rdf:resource="#Device"/>
     <rdfs:subClassOf>
      <owl:Restriction>
       <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
       >1</owl:maxCardinality>
       <owl:onProperty>
        <owl:DatatypeProperty rdf:ID="hasOperatingSystem"/>
       </owl:onProperty>
```

```
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
     <owl:Restriction>
      <owl:onProperty>
       <owl:DatatypeProperty rdf:ID="hasRAM"/>
      </owl:onProperty>
      <owl:maxCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:maxCardinality>
     </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="VideoCommunicationDevice">
   <rdfs:subClassOf rdf:resource="#CommunicationsDevice"/>
  </owl:Class>
  <owl:Class rdf:ID="MotionSensorDevice">
   <rdfs:subClassOf rdf:resource="#SensorDevice"/>
  </owl:Class>
  <owl:Class rdf:ID="MobileTelephone">
   <rdfs:subClassOf rdf:resource="#Telephone"/>
  </owl:Class>
  <owl:Class rdf:ID="AudioSensorDevice">
   <rdfs:subClassOf rdf:resource="#SensorDevice"/>
  </owl:Class>
  <owl:ObjectProperty rdf:about="#hasAllowedUsers">
   <rdfs:domain rdf:resource="#Device"/>
   <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1#Person"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#hasLocation">
   <rdfs:domain rdf:resource="#Device"/>
   <rdfs:range
rdf:resource="http://pervasive.semanticweb.org/ont/2004/06/geo-
measurement#LocationCoordinates"/>
  </owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="hasProtocol">
  <rdfs:range rdf:resource="#Protocol"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#hasStandard">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Protocol"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasBandwidth">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Protocol"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasMessageSchema">
  <rdfs:domain rdf:resource="#Device"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#anyURI"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasOperatingSystem">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#ComputingDevice"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasPhoneNumber">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#Telephone"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasInstalledApplications">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#ComputingDevice"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:about="#hasRAM">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
  <rdfs:domain rdf:resource="#ComputingDevice"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="hasName">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Protocol"/>
```

```
  </owl:DatatypeProperty>
  <geom:LocationCoordinates rdf:ID="loc_laptop2">
   <geom:altitude rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >-64</geom:altitude>
   <geom:latitude rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >-48</geom:latitude>
   <geom:longitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >144</geom:longitude>
  </geom:LocationCoordinates>
  <geom:LocationCoordinates rdf:ID="loc_desktop2">
   <geom:longitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >64</geom:longitude>
   <geom:latitude rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >96</geom:latitude>
   <geom:altitude rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >-96</geom:altitude>
  </geom:LocationCoordinates>
  <LaptopComputingDevice rdf:ID="laptop1">
   <hasLocation>
    <geom:LocationCoordinates rdf:ID="loc_laptop1">
     <geom:latitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
     >128</geom:latitude>
     <geom:longitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
     >32</geom:longitude>
     <geom:altitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
     >-224</geom:altitude>
    </geom:LocationCoordinates>
   </hasLocation>
  </LaptopComputingDevice>
  <per:Person rdf:ID="Smith_John">
```

```
    <per:familyName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Smith</per:familyName>
    <per:gender rdf:resource="#male"/>
    <per:firstName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >John</per:firstName>
    <per:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    >1979-02-17</per:birthDate>
  </per:Person>
  <DesktopComputerDevice rdf:ID="desktop2">
    <hasLocation rdf:resource="#loc_desktop2"/>
  </DesktopComputerDevice>
  <FixedLineTelephone rdf:ID="telephone1">
    <hasLocation>
      <geom:LocationCoordinates rdf:ID="loc_telephone1">
        <geom:altitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >-96</geom:altitude>
        <geom:longitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >72</geom:longitude>
        <geom:latitude
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >48</geom:latitude>
      </geom:LocationCoordinates>
    </hasLocation>
  </FixedLineTelephone>
  <LaptopComputingDevice rdf:ID="laptop2">
    <hasLocation rdf:resource="#loc_laptop2"/>
  </LaptopComputingDevice>
  <Wireless rdf:ID="WiFi802.11g">
    <hasStandard rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >802.11g</hasStandard>
  </Wireless>
```

```
  <Wireless rdf:ID="WiFi802.11b">
    <hasStandard rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >802.11b</hasStandard>
  </Wireless>
  <per:Person rdf:ID="Kelly_Joe">
    <per:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    >2005-02-17</per:birthDate>
    <per:firstName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Joe</per:firstName>
    <per:gender
rdf:resource="http://pervasive.semanticweb.org/ont/2004/06/person#Male"/>
    <per:familyName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Kelly</per:familyName>
  </per:Person>
  <per:Person rdf:ID="Murphy_Mary">
    <per:firstName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mary</per:firstName>
    <per:gender rdf:resource="#female"/>
    <per:familyName
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Murphy</per:familyName>
    <per:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    >1945-02-17</per:birthDate>
  </per:Person>
  <Wired rdf:ID="Ethernet">
    <hasStandard rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >802.3</hasStandard>
  </Wired>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 1.3, Build 227)
http://protege.stanford.edu -->
```