Designing Telecommunication Service Management Systems

A thesis submitted to

University of Dublin, Trinity College

for the degree of

Doctor of Philosophy.

Vincent P. Wade

Department of Computer Science

Trinity College

Dublin

May 2006

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.

Vincent P. Wade

May 2006

Permission to lend or copy

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

Vincent P. Wade

May 2006

Acknowledgements

I would like to thank my supervisor, Professor Jane Grimson, for all her support and advice throughout this research. I would also like to thank my colleagues and friends in the Knowledge and Data Engineering Research Group (both past and present) for making the group such a stimulating and enjoyable place in which to research.

I would like to especially thank my wife Sandra, whose constant support, generous encouragement and faith helped ensure the success of this thesis. Thanks also to our children Andrew and Rachel, whose arrival while this research was being carried out, helped keep us all balanced between work and family life.

Finally, I would like to express my deep gratitude to my parents for instilling in me an interest in learning and an appetite for knowledge. Thanks also to my sister and brothers for their encouragement in completing this research.

Abstract

In today's telecommunication industry, the management of services is seen as a key enabler and differentiating factor in the flexible delivery of telecommunication solutions. Increased competition and globalisation of markets have created unique pressure on telecommunications service providers to support service oriented, reusable management components which can optimise the provisioning and adaptation of business and systems processes. This thesis proposes a model driven development strategy which provides integrated, but separate support for the development of reusable component designs as well as the flexible development of business process driven systems constructed using these reusable component designs. The thesis specifies two integrated development guidelines which comprise the methodology. The thesis also describes the validation and evaluation of the guidelines in two trials. These trials involve the development of a catalogue of component designs and the development of several different telecommunication management solutions constructed using these component designs. The thesis also provides a comparison of the proposed guidelines with existing telecommunication development processes and mainstream software industry development processes.

Table of Contents

1 INTRODUCTION

"A key goal of integrated management is to encompass the entire information technology system up to and including the business processes themselves." (Calo & Kung IM2005 'Managing New Networked Worlds' IM2005)

Telecommunications Service Management is a key enabler for the rapid introduction and flexible delivery of telecommunication services and applications [TeleManagement Forum 2005]. The last decade has seen the continued evolution of management development techniques and service architectures, so as to improve development, reuse and re-configuration. Telecommunication management methodologies have tended to focus on particular standards (M3100) or specific technology based solutions e.g. EJB, .NET, or have suggested general software development approaches without specific, relevance to telecommunications management challenges.

The key problems facing development methodologies for next generation telecommunications are the need to reduce the complexity of the development effort, both in the development of service/components as well as the integration of these components into management applications; the need to support possible automation in the development process (and hence increase the speed with which management services and applications can be developed); and the need to make best use of mainstream, technology neutral, development techniques (e.g. UML, MDA) whilst supporting the appropriate application of emergent standards e.g. TMF, DMTF.

This thesis focuses on the methodologies needed to provide flexible and dynamic development of telecommunication service management systems. This chapter first outlines the scope of telecommunication service management, as defined by several standards bodies and industrial fora. The chapter then sets the context for the thesis, by identifying the main drivers for telecommunication management and introducing the key management standard bodies and industry consortia in the area. The chapter presents the motivation and objectives of the thesis and provides an overview of the technical approach for the thesis's research. Finally the overall structure of the thesis is presented.

1.1 Telecommunication Service Management

Within the evolving telecommunications management landscape, this thesis concentrates on the modelling of telecommunication service management components and systems. Service management is concerned with the management of telecommunication services (and value added services) above the network management level. Several definitions for service management have been proposed. Telecommunication Service Management[1] typically focuses on the delivery of a service and involves such management functionality as: order management, inventory management, provisioning, activation, network topology management and maintenance, and stability/performance diagnostics of telecommunications service and their networks (Wikapedia 2006).

In the International Telecommunications Union (ITU) Telecommunication Management Network (TMN) Standard, service management is concerned with those aspects that may be directly observed by the users of the telecommunication network. These users may be end-users (customers) or other service providers. Examples of such service management functions would be quality of service management (delay, loss etc), accounting, subscription management, service provisioning etc. In TMN, service management is one of five management layers which logically divide the operation and management of the network and the services which operate upon it [Pras 1999]. These layers consist of:

(a) Network Elements or nodes in the network

(b) Network Element Management Layer which is responsible for the management of individual network nodes

(c) Network Management Layer which is reponsible for managing the overall network

[1] Telecommunication Service Management Systems typically include such functionality as billing, charging, accounting, (quality of service) assurance, fault detection and prevention, etc.

(d) Service Management, which is responsible for managing services across the network, and

(e) Business Management which focuses on the business related activities of operating networks and services. Figure 1.1 illustrates the cone or pyramid logical model for management abstraction.

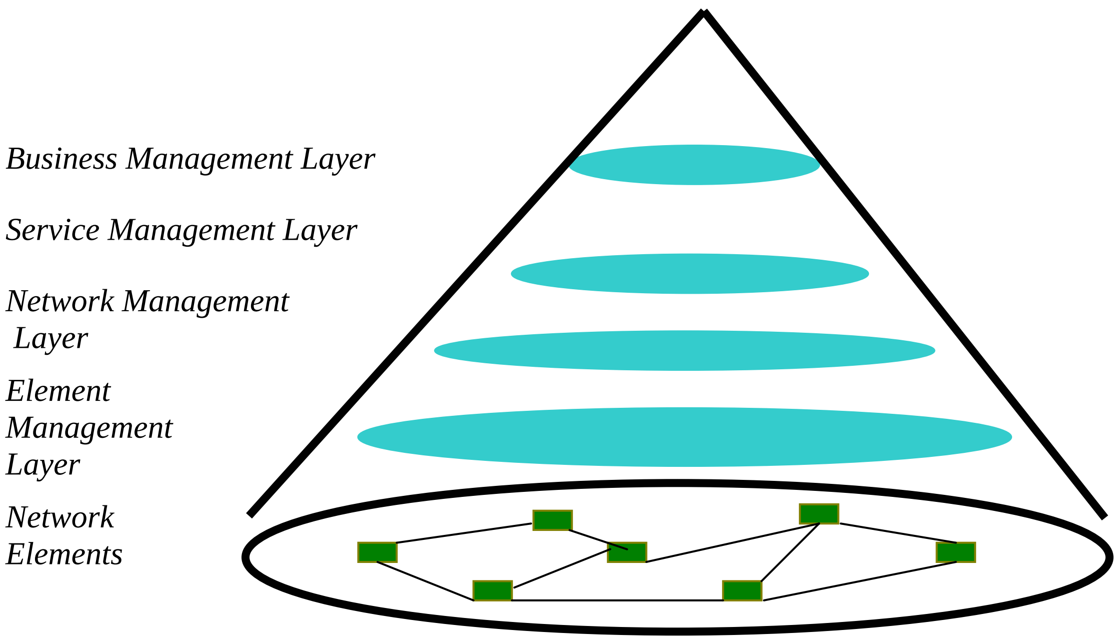


Figure 1.1 ITU Telecommunications Management Network, Logical Layered Architecture [ITU-T 2000]

The TeleManagement Forum takes a slightly wider interpretation of the scope of service management (than the TMN standard), dividing service management into several vertical process groupings, in areas including Operations Support & Readiness, Service Fulfilment, Service Assurance and Service Billing, as shown in Figure 1.2 [TeleManagement Forum eTOM 2005]. It identifies a non-exhaustive set of service management processes including service configuration and activation, service problem or fault management, quality of service management and service rating or accounting management. The TeleManagement Forum also identifies a slightly different hierarchical layering than in TMN standard. At the top level is Customer Relationship management, which offers business process interfaces directly to the customers of the telecommunication service provider. Below the service management layer, the TeleManagement Forum depicts Resource Management (which includes processes to management application resources, Computing and Network resources), and Supplier/ Partner Relationship Management.

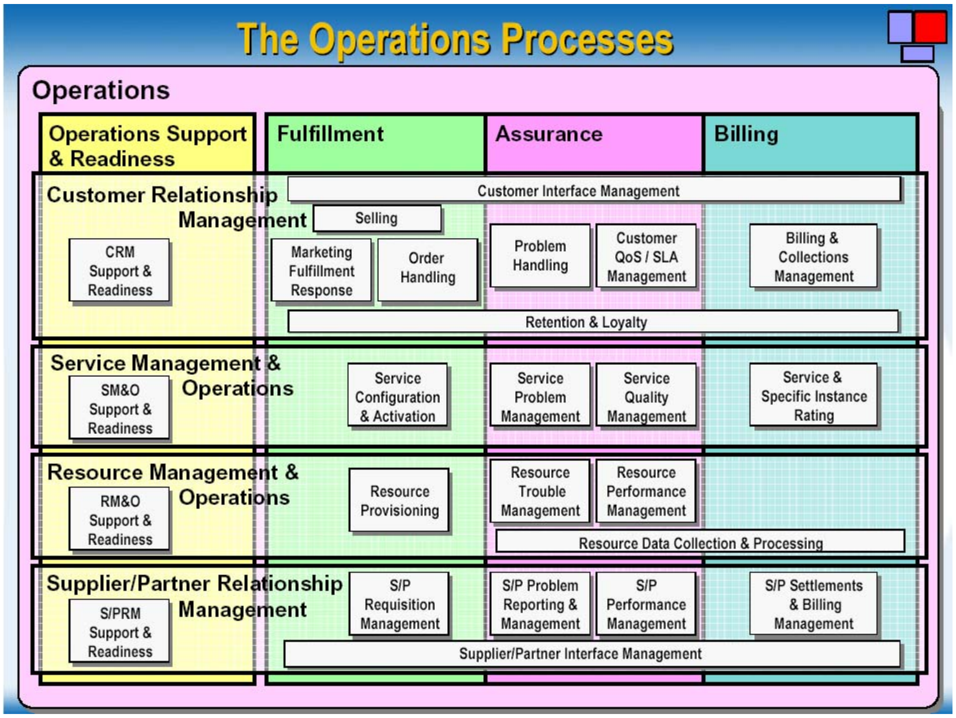


Figure 1.2 TeleManagement Forum Enhanced Telecommunication Operations Map – Operations Processes, reproduced from TeleManagement Forum extended Telecommunications Operations Map [TeleManagement Forum 2005]

1.2 Key Drivers for enhancing Development Approaches for Telecommunication Management

This section first highlights some of the key drivers which are influencing development approaches for telecommunications management. These drivers set the context within which management systems have to be developed and identify important challenges that development methodologies for service management must address.

Driver #1: Deregularization and Globalization

As a result of deregularization of the telecommunication industry in Europe and US in the nineties and the ensuing globalization of telecommunication industry, there has been a rapid growth in both the number and sophistication of telecommunication services being offered in today's telecommunications service market [Goldman 2004]. These services range from (managed) virtual private networks to value added services such as distributed enterprise applications, multimedia entertainment services, and business-to-business application integration services [Adams 1996], [Schulzrinne 2006]. Because of the strength of competition, telecommunication providers are under increasing pressure to accelerate the deployment and provisioning of services. Rapid, flexible design and the automation of telecommunication service and network management are seen as a key competitive discriminator between providers [Wade 2000]. However, this marketplace has become even more competitive as the industry climbs out of the 'dot com' crash in the early part of 2000s. This has seen increasing market share being taken by new telecom providers as well as existing telecom providers merging or being acquired [FCC 2005].

Network and Telecommunications Service Management systems are being seen as a key enabler of telecommunications providers to be more flexible and cost efficient [TeleManagement Forum 2005]. Traditionally such management systems were developed by or on behalf of the telecommunication provider. However, such bespoke management systems tended to result in very high costs as well as spiralling (systems) complexity. In an effort to reduce costs and manage complexity, telecommunications providers have increasingly adopted a strategy of purchasing individual management systems. However, such individual systems exhibit significant difficulties in interoperability and functional reuse. Telecommunication providers are migrating some of their management functions to 'off the shelf' applications to satisfy their management requirements [TeleManagement Forum NGOSS 2005]. Thus the trend for 'off the shelf' purchasing of management functionality (components) is driving (a) the development of telecommunication management architectures (for both networks and the services which run upon them), (b) the technologies being researched and applied within the management, and (c) the business models, and processes, which ultimately the management systems must support.

Paradoxically, deregulation of the telecommunication industry has also led to increased co-operation between telecom service providers. Such co-operation is generally aimed at gaining access to global markets. Therefore another driving force for such co-operation is the requirement of interoperability and fair-trading amongst existing and newer providers and service providers e.g. Federal Communications Committee rules in USA. The requirements exerted by this increased competition between providers and the (paradoxical) co-operation of providers are greatly affecting the way telecommunication services are managed today and will be managed in the future. This need for greater interoperability across organisational boundaries, can also been seen as a consequence of the globalisation of telecommunication services, where global service delivery usually requires significant management system interactions across different telecommunication service providers. An example of such multi-telecommunication provider co-operation would be in the provisioning of Virtual Private Networks across several organisations and continent boundaries [Wade 2002, Lewis1999a]. Because of the ever-increasing business-to-business computing integration on the Internet and the growth of global markets, the need for inter-telecommunication provider co-operation is ever increasing.

Driver #2: Standardization within Telecommunications Operational Support Systems

Since the 1980s standardization has been a key influence for the design and integration of telecommunication management systems. The architectural landscape of management systems include the standardization effort of several telecommunication groups e.g. the International Standards Organisation (ISO) Study Group 4 on Telecommunication Network Management (TMN) [ITU-T 2000] and the Internet Engineering Technology Taskforce [IETF]. However as telecommunication becomes embedded into the fabric of modern organisations, mainstream computer software industry fora are becoming more influential in the telecommunication management domain. Examples of such computer industry initiatives within the telecom management domain include the TeleManagement Forum [TeleManagement Forum] and Distributed Management Task Force (DMTF) [DMTF].

Thus, one of the problems with the development of management systems is that they frequently need to adopt or are influenced by several standards, rather than one single standard. This 'multi standard framework' problem can be illustrated in the TeleManagement Forum's management process areas (e.g. Fulfilment, Assurance and Billing) [TeleManagement Forum eTOM 2005]. Frequently, several standards (and their associated information models and protocols) would be relevant to a management application area such as telecommunication accounting applications or telecommunication performance applications. For example, accounting management may use emergent Internet Protocol Data Record information models [IPDR 2002] and its specification for the representation of accounting information, but may also need to be consistent with IEFT/DMTF or TMN standards for network and service modelling.

Driver #3: Adoption of mainstream IT Middleware

Another significant factor which is influencing telecommunication service developers and management providers is the drive to use more mainstream information technology solutions and techniques rather than, relying on telecommunication industry specific technologies and techniques. The rationale is that the mainstream information technology industry is far more widespread than that of telecommunication management and has far larger research investment, development and deployment. Therefore, adoption of such technology could offer much greater scope for cost reduction. A second key benefit of mainstream IT middleware is the potential to provide greater reuse of applications or components. Such middleware provides significant services to ease the reuse of software, supporting such activities as identification of distributed functions/components, integration of components etc. Thus, several telecommunication management industrial fora (e.g. TeleManagement Forum and DMTF) are attempting to fuse and enhance mainstream computing and telecommunication technologies to support the development and deployment of telecommunication management solutions [Wade 2000].

However, one of the difficulties in adopting mainstream technologies, especially middleware technology, is the relatively rapid rate of revolution within these technologies. In the last eight years alone, four different middleware initiatives have been embraced by the telecommunication management vendors, namely; OMGs CORBA, SUNs Enterprise Java Bean middleware, the Microsoft's .NET initiative and the W3Cs XML family of technologies including SOAP and Web Services. The latest architectural development in IT middleware has been the drive toward Service Oriented Architecture. Here the reuse is based on services offered anywhere within the network. The most common example of such Service Oriented Architectures is that of W3C's Web Service Architecture [WC3 2004].

Although these middleware technologies are in some ways comparable, development of solutions based on each, tend to have different design approaches and conventions. Porting applications and components across these infrastructures can involve significant effort both in porting the management applications and components and in re-design of these applications and components to make best use of the new middleware. Such 'technology churn' is worrying for telecommunication service providers as their real goal is the business logic and processing which is wrapped within or built on top of such an infrastructure, and is typically not the distributed processing infrastructure itself. Although each infrastructure offers different benefits and savings for the management system or component developer (e.g. location and distributed transparency, naming, remote invocation, persistence, queuing etc.), having to perform significant re-engineering to adapt to a new infrastructure as the technologies churn, needs to be carefully addressed. Therefore such middleware technology churn needs to be mitigated by the development processes and strategies adopted by the telecommunication management vendors and providers.

Impact of these drivers on methodologies for telecommunications management systems development

It is possible to derive important challenges which design and development approaches need to address, if the systems they realize are to be capable of success in the telecommunications market of today and tomorrow. Thus, development methodologies for telecommunications management systems need to (Wade 2000):

(i) Assist the flexible integration of management components and systems, (both within a telecommunication service provider and between telecommunication service providers). (Driver #1)

(ii) Reduce the complexity of developing management systems, increase the potential for reuse of management functionality and increase the speed of development and deployment of these systems (Driver #2)

(iii) Reduce the costs of developing management systems (Driver #1, Driver #2)

(iv) Increase the level of automation of management system to provide greater capability and to manage higher levels of complexity in networks and systems (Driver #1, Driver #2)

(v) Adopt development strategies which are capable of working with a range of the *de-jure & de-facto* telecommunication management standards ITU-T, IETF, DMTF and TeleManagement Forum (Driver #2)

(vi) Harness more mainstream information technologies and development techniques rather than maintain a reliance on telecommunication specific technologies. (Driver #3)

(vii) Ensure design level independence of the business logic embedded in the telecommunication management components and systems, and the distributed middleware upon which such components and systems operate. (Driver #3, Driver #2)

1.3 Thesis Objectives

This thesis addresses the key challenge of (a) how to develop reusable designs for service management components and (b) how to the develop service management systems whose construction can be based on these (component) designs. In particular, the thesis proposes development activities for the design of reusable components and for their integration using a business process driven approach. The key benefit of the approach is to reduce the complexity of the traditional development processes and to provide focused, easy to apply guidelines for the specific goals of component design reuse, and business process driven development. The thesis identifies the key design activities and design workflows needed to model both reusable components and business process driven systems (which utilise these reusable components) for telecommunciation service management.

The thesis proposes an integrated development methodology which addresses the key motivations for service management component developers and service integrators. More specifically the thesis

(i) researches the development approaches for telecommunications service management

(ii) proposes and develops a methodology which provides an integrated approach for (a) the *modelling of (reusable) telecommunciation service management componentware (called building blocks)*, and (b) the development of *business process driven approach to* telecommunication management systems which supports the reuse of management component designs (building blocks).

(iii) validates the methodology with the development of a framework of telecommunications management building blocks and telecommunication systems across a range of management application areas (namely fulfilment, assurance and billing).

The resultant methodology is particularly focused on meeting the requirements of a set of telecommunciations service providers, and embeds appropriate reference to management standards, industry standards as well as methodological best practice. The benefits of the methodology is that it reduces the complexity of developing such components and systems, and promotes the reuse of component designs. It also supports the business process approach to systems construction, which is highly popular in telecommunciation management systems.

This MOdel Driven Development methodology (called MODD) has evolved over three European research projects [Prospect, FlowThru, FORM]. In particular, it has been applied and evaluated within the FORM European Research project, funded under the IST Research Programme [FORM 2002].

The methodology was trialled and evaluated during the design and implementation of telecommunication service management components and systems across a range of management application areas, such as service accounting, fulfilment and assurance. The systems and components designed using the methodology were implemented using different technologies e.g. Enterprise Java Beans, OMG CORBA and Workflow based systems. The management systems developed involved multiple telecommunication management standards (e.g.. IETF SNMP MIBs, DMTF CIM, TeleManagement Forum's SID, IETF IPSec). The approach proposed in the thesis seeks to enable open interface specifications to facilitate flexible component integration and enhance mainstream information systems development approaches for telecommunication management. The methodology supports a model driven approach to component and system development. This approach supports the re-use of the management component designs and their implementation across different middleware technologies. The methodology and the description of the systems designed using the methodology have been peer reviewed in international conferences (IEEE NOMS, IEEE IM), have been the subject of tutorials for research and industry at international conferences (IEEE NOMS 2002), have been presented to industry bodies e.g. TeleManagement Forum NGOSS development team and have been reviewed very favourably by an international review panel for European Research (FORM Evaluation Panel 2001 & 2002).

1.4 Contribution of Thesis

The contribution of the thesis is in the design and evaluation of the two integrated guidelines which provide lightweight, easy to follow methodologies for developing (i) component designs and contract interfaces, (ii) business process driven systems development which facilitate the reuse and integration of pre-existing interface contracts and component designs. Traditional development approaches focus on one or other of these approaches, e.g. the Rational Unified Process focuses on class reuse and the design of a system as a collection of classes at runtime. Typical business process driven approaches seek to develop sequences of business activities, but provide little guidance as to their implementation using reusable component designs. The MODD Methodology harmonises the two approaches to provide business process driven systems which can be constructed using reusable component designs (which we term Building Blocks). The guidelines are developed to support two key telecommunications management stakeholders, namely service management component vendors and telecommunications management system integrators.

A second aspect of this contribution is the tailoring of MODD to address the specific needs and challenges of the telecommunication service management e.g. ability to use multiple standards and to support for the ‘business model’ specifics of the telecommunications management environment. The final aspect of the thesis’ contribution is that modern OO model driven methodology can be very complex and difficult to adhere to [Beck 2001]. MODD is designed as a lightweight, agile development process which demonstrates high usability for telecommunication management development.

The MODD methodology considerably reduces the complexity of general software development processes e.g. RUP, by providing tailored, lightweight processes that focus on the needs of the various stakeholders in the telecommunication value chain. By dividing the development processes into two complementary guidelines, this simplification and fitness-for-purpose is further enhanced. The guidelines also promote reuse and the construction of management systems using predefined building blocks (component designs). Finally, the guidelines are defined and make use of UML notations so that existing UML based development tools can be employed to implement the development processes.

1.4.1 Publications from Thesis

During the development of the thesis, the research was peer reviewed and published in several international research conferences and journals. The publications which are based on the research in this thesis are:

1. “Flexible Automated Enactment of Process Driven Telecommunication Management”, Wade V, Muldowney S, Fuller J, ‘Interoperable Communications Networks Journal’ - Advanced Strategies and Technologies for Broadband Telecommunications Management’, Vol 1/2, Baltzer Scientific Publishers Neitherlands ISSN 1385 9501, pages 49-61
2. ‘Three Keys to Developing and Integrating Telecommunications Service Management Systems”, V Wade, D Lewis, IEEE Network, Volume 37, Issue 5, May 1999 Page(s):140 - 146
3. “Service Management and the Telecommunications Information Networking Architecture”, V Wade, Computer Communications Journal, Elsevier, Vol 22, Number 18, Dec 1999, ISSN 0140-3664, pages 1633-1638
4. “Workflow, A Unifying Technology for Operational Support Systems”, V Wade, T Richardson, IFIP/IEEE International Symposium on Networks and Operations Management Systems NOMS, April, Hawaii, USA 2000, Page(s):231 - 246
5. “Integration approaches to Telecommunications Management Systems”, Vincent P. Wade, D. Lewis, C. Malbon, T. Richardson, L Sorensen, C. Statopoulos “7th International Conference on Intelligence and Services in Networks, IS&N 2000”, LNCS 1774, Springer Verlag, Athens, Greece, Feb 2000, pages 315-332
6. ‘Towards a Framework for Management Business-to-Business eCommerce Chain”, Vincent Wade, David Lewis, Jacque Brooke, William Donnelly, Chapter VI in ‘Challenges Managing Virtual Web Organisations in the 21st Century: Issues and Challenges’ Ed. Ulrich Franke, Idea Group Publishing, 2002, pages 107-118
7. “A Model-Driven Approach to Component Based Management, Tutorial T9”, Vincent Wade, David Lewis, IFIP/IEEE 8th International Symposium on Networks and Operations Management Systems NOMS, Florence, Italy, April 2002

1.5 Thesis Overview and Technical Approach

This thesis has been developed within the context of evolving telecommunication service management architectures and business drivers. Chapter 2 explores the different actors in telecommunication management development and operation, namely standardization bodies, management component developers, system developers/integrators and service providers. It outlines their roles and influences within the overall context of telecommunication management system development. The thesis focus is on two such actors, namely system integrators and management components vendors (developers). The key requirements for the providers of telecommunication management systems are identified and the challenges of integrated component and system development are explored. The contribution of the main telecommunication management standards architectures with regard to their support for service management is analysed. This is important as, given the wide acceptance of the architectures and their existing deployment within the telecommunication service provider’s infrastructure, the proposed methodology should be applicable to one or more of these architectures.

Chapter 3 analyses the development methodologies, which have arisen both within the telecommunication community or have been adopted within the community from mainstream Information Technology/Software Engineering. It identifies the trends in the evolution of these two steams of management development methodologies. It also briefly highlights the technology, which was used to integrate the components modeled using the methodology.

Chapter 4 proposes the MODD methodology which consists of two development guidelines. The first is a development guideline for the design of management components. The second is a (management) process oriented development guideline for telecommunication service management systems, which faciliates construction using management components developed in the first guideline. The main aim of this chapter is to present the component based modelling guideline and illustrate its application using an example based on service monitoring. This guideline is focused on the analysis and design stages of development rather than implementation issues. Underpinning the methodology are two key conceptual notions of a Building Block and Building Block

Contract. Stated simply, a Building Block is the design model(s) which describe a component[1]. A Building Block Contract is an interoperability specification which provides the only means of interacting with a Building Block. A Building Block Contract specification contains a grouping of interface signatures, information models, and interaction behaviours, which can be reused to support telecommunication management business processes. A Building Block can support (implement) one or more Building Block Contracts. The chapter describes the Building Block development guideline and illustrates the design of Building Block Contracts.

Chapter 5 presents a business process driven guideline for developing management systems. This guideline offers a unified means of designing service management business processes, mapping the processes onto existing or new management Building Block Contracts and co-ordinating the execution of these components to control flow across these processes.

In order to evaluate and validate the methodology, it was applied within the FORM EU IST Project. This involved the design and implementation of a set of components (which at design time are called Building Blocks) across a range of management functional areas. The project implemented several management processes in contrasting management areas, based loosely on the TeleManagement Forum Business processes, using a variety of implementation technologies. Chapter 6 describes the evaluation of the guidelines based on the findings of the trials and validations, as well as on a qualitative review of related work in the research area.

Finally based on the thesis's original goals, Chapter 7 draws conclusions on the level of achievement. The chapter also presents conclusions based on the evaluations and development experiences. The chapter concludes with an indication as to how the methodology could be improved and extended in the future.

[1] A component is an atomic unit of deployment of a reusable piece of software. (Kruchten 2000)

2 TELECOMMUNICATION MANAGEMENT – ARCHITECTURES AND INFRASTRUCTURES

2.1 Introduction

The Telecommunication Management domain is littered with management architectures and *de-jure* or *de-facto* standards initiatives e.g. IEFT, ITU-T, TINA, TeleManagement Forum and DMTF. This chapter first identifies the key actors involved in the telecommunication service management community. These include standardization bodies, management component developers, system developers/integrators, service providers. The chapter uses this categorization to outline different roles and influences within the overall context of telecommunication management system development. Current telecommunication management architectures, which need to be considered when developing telecommunication service management, are then examined. Finally, the chapter identifies different approaches to support the integration of service management systems. These approaches are used later to integrate the component designs and systems design using the thesis' methodology, MODD.

2.2 Relationships between Telecommunication Service Management Community in the Telecommunication Market Place

The actors in the telecommunication service market place can be characterised as a set of independent standards bodies, independent software vendors, system developers/integrators, service providers and service customers [Lewis 99].

Since the 1980s, the telecommunication industry has been greatly influenced by standardization. This was seen as vital to avoid 'network vendor lock-in', where telecommunication providers were forced to use the network equipment vendor's proprietary management system to manage that vendor's communications devices. The use of such proprietary management solutions was prevalent up to the late 1980s and early 1990s when standards based management products began to emerge. In this period the key standard bodies were OSI (with the X700 standards series of specifications and the M3000 series specifications) and the IETF (with the SNMP family of network management standards). The mid and late 1990s also saw the emergence of industry based consortia guidelines/standards, e.g. Network Management Forum (subsequently renamed the TeleManagement Forum), Distributed Management Task Force (DMTF), Telecommunication Information Network Architecture Consortium (TINA-C) etc. These industry bodies have been predominantly influenced by a particular market segment viewpoints; TeleManagement Forum being heavily influenced by Telecommunication Providers, where as DMTF being principally driven by computer industry and enterprise network vendors. Therefore it is not surprising to note that the TeleManagement Forum originally adopted OSI based approaches and latterly Internet based approaches[1] whereas the DMTF was more centered on enterprise technology e.g. distributed object systems' technology such as COM, and mainstream computer industry technologies such as WWW, XML, and UML.

The early 2000s have seen a steady convergence of the enterprise management and telecommunication management markets. This has occurred for many different reasons; the deregulation of the telecommunication markets; the globalisation of markets; the increasingly common distribution of enterprises over wider areas; the dominance of IP as a common data communication protocols for wide area as well as local area networking; the significant increase and reliance on enterprise-to-enterprise (so called Business-to-Business) e-commerce; the increasing drive to lower operational costs and increasing software reuse. This has meant that telecom management industry standard bodies such as DMTF and TeleManagement Forum are increasingly adopting mainstream computer

[1] Since the demise of the use of key OSI management protocols, the TeleManagement Forum has attempted to develop designs which are independent of specific technology (so called technology neutral architectures).

industry technologies for systems specification and implementation rather than specifying telecommunication specific technology.

From a management design perspective, both the TeleManagement Forum and DMTF are adopting the Unified Modelling Language to represent their standards models. Thus the current standards/guidelines from the TeleManagement Forum describe frameworks and architectures using UML and various specifications languages, while DMTF's Web Based Enterprise Management architecture and Common Information Model is being represented in UML.

The interrelationship between the telecommunications management standards, management component vendors system integrators and service providers is presented in Figure 2.1. Management component vendors are being heavily influenced by the needs of the telecommunications providers, and the evolving technical standards in the component vendor's target management area. These standards focus on interface, protocols and Management Information Base (MIB) specifications, both for generic (core) aspects of management as well as aspects specific to the type of management under development e.g accounting management standards, fault management standards. The standards also relate to the type of network, network device, or system being managed e.g. specific management standards associated with managing sensor type networks, wireless networks and fixed networks. The intention is that the standards provide for greater interoperability of the component vendors software. This potential interoperability is key for system integrators who use the standards and interfaces to design management solutions (typically) across multiple management component vendors.

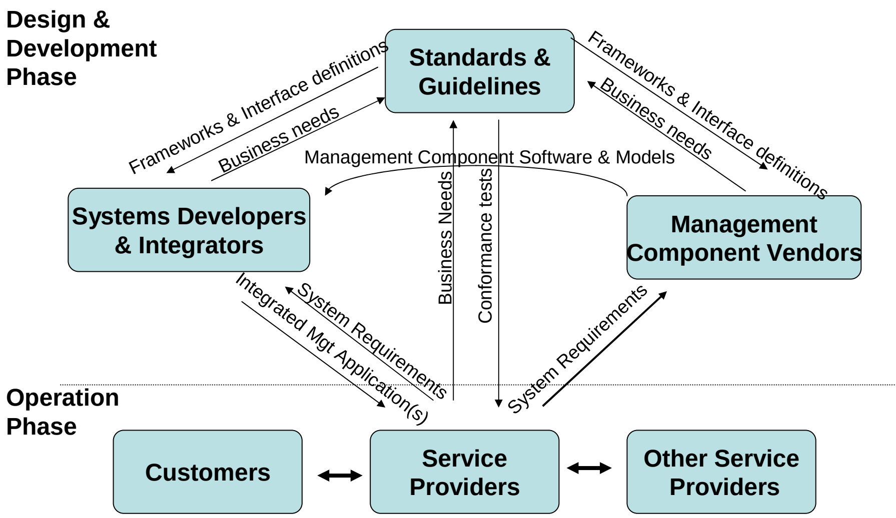


Figure 2.1: Market context for the development of Management Systems (based on [Lewis 99]

Service Providers are deploying the 'solutions' developed by the system integrators to operate the required management solutions for their specific service offerings. This is typically the case nowadays as the complexity and scale of the integration of the management software is beyond the expertise of the service provider who is more focused on the actual lifecycle management of the service and the deployment of new services rather than the development of management software[1]. However the relationship between system integrators (system or solution developers) and the service

[1] Some large Service Providers still perform full systems integration design and implementation but this is much less common and typically still involves some separate systems integration consultancy

provider is typically very close during development and deployment projects[1]. To reduce the complexity and cost of such integration, service providers are attempting to reduce the number and variety of management technology platforms in use within their networks.

2.3 Drivers for Telecommunication Management Services and Systems

In the liberalised US and European telecommunication markets, the integration of management functionality is a critical property of operational support systems. The scale, heterogeneity and geographic distribution of network and service management systems across the value chain of network and service providers, presents great difficulties in the fulfilment, assurance and billing of new services [Wade 2002a]. For example, the TeleManagement Forum conducted a survey involving the principal Public Network Providers in Europe and identified the main business drivers for these service providers as the need to:

- Reduce costs in provision of services;
- Improve process flow across service provider organisation (and management systems);
- Increase management process automation;
- Increase customer management control;
- Improve quality of service management; and
- Increase the range of management services;

Several trends have emerged as a result of these drivers. The first can be seen from the general acceptance that bespoke management solutions are no longer tenable. There is a much greater willingness of telecom providers to outsource or purchase ‘off the shelf’

[1] This use of system integrators for solution development and deployment is common in the software industry in general, but within the Telecommunication Management industry, the adherance to *de-jure*

telecommunication management solutions and components [Wade 1999]. A very important element in this approach is the significance placed on ease of integration of procured management components [Adams 1996]. In particular, telecommunication providers (or system integrators on their behalf) need to create operational support system solutions from reusable telecommunication management components that may be drawn from multiple origins. Thus these practitioners need to be able to make reasoned selections from existing solutions (standardised or otherwise) while ensuring the integrity of the information flows required in achieving business and operating requirements. Thus service providers, system integrators and system developers need an open market for reusable management components, which allow the building of software systems from reusable components. Much of the more recent effort of standardization work can be seen as an attempt to stimulate and foster this open market for management componentware.

2.4 Standards based Architectures for Integrated Service Management

There are several important architectures which have been widely accepted and adopted, and which have had or are having considerable influence in the engineering of service management. These standard architectures have originated from both formal internationally accredited standardization bodies (e.g. ITU-T) as well as industrial fora e.g. TeleManagement Forum. This section outlines some of the principal architectures which support service management. More specifically, the section identifies the key influences and contributions of these architectures to the design of modern service management systems. The survey includes the Telecommunication Information Network Architecture Consortium's Service Architecture; TeleManagement Forum's Extended Telecommunication Operations Map (eTOM) and Shared Information/Data Model (SID) ; and Distributed Management Task Force's Web Based Enterprise Management (WBEM) and Common Information Model (CIM) approaches. In the interests of brevity and focus, two older management architectures, i.e. those based on the SNMP and CMIP

and industry technical specification and guidelines is much more significant.

manager/agent paradigm, are not included as they are primarily used for low level network resource management. However, several of the standards suveyed in this section, allow for integration with such protocols e.g. TeleManagement Forum, DMTF.

2.4.1 Service Management in Telecommunication Information Networking Architecture (TINA)

The Telecommunication Information Network Architecture (TINA) Consortium [TINAC] was an international collaboration which aimed at defining and validating an open architecture for telecommunication systems for the broadband, multimedia and information era. Its members included a large number of network providers, computer vendors and telecommunication manufacturers. The consortium had three goals:

(i) to facilitate versatile multimedia and information services

(ii) to ease the creation of new service and management of these services and the networks over which they operate

(iii) to create an open telecommunication and information software component marketplace

An innovative aspect of TINA's approach, as compared to the approach of other management standards at the time, was the adoption of a combination of main stream, distributed object oriented computing middleware, with concepts and standards drawn from the telecommunication and computing industries. TINA defined an overall software framework for telecommunication systems. This framework attempted to address the needs of traditional voice-based service, future interactive multimedia services and information services. The TINA initiative has now completed, but it was very influential in both the research of telecommunications management systems and in the industrial fora.

2.4.1.1 TINA Business Model

One of the critical contributions of the TINA architecture was its attention to the business or enterprise model. The TINA business model identified the various 'roles' involved when considering a virtual marketplace for services and networks (as previously depicted in Figure 2.1). Technology, economics and regulatory boundaries determine the separation between these business roles. Entities, individuals or legal entities, that take on a TINA business role are referred to as *stakeholders*. TINA attempted to standardise the relationships and interfaces between the different business roles, referred to as TINA Reference Points (RP). Every RP where a *user–provider* relationship can be identified between the business roles, consists of both an *access* and a *usage* part. The access part defines the procedures required to set up a trusted relationship, an *access session*, between two stakeholders. The usage part details the mechanisms available to the user business role for actually using the services provided by the provider business role. In the context of TINA, a service is be interpreted as a collection of interface operations offered by a server.

The TINA Business Model, in Figure 2.2, identifies five stakeholders namely, *Consumer*, *Retailer*, *Broker*, *3rd Party Service Provider*, *Connectivity Provider*. A *Consumer* is a business role representing the end-user in a TINA system, in the sense that s/he has no interest in generating revenue by being engaged in the deployment of a TINA system as a specific technology. Consumer stakeholders can equally represent large corporate networks or individual terminals. They establish contractual relationships with *Retailer* stakeholders, which represent a "one-stop shop" for TINA services for Consumers.

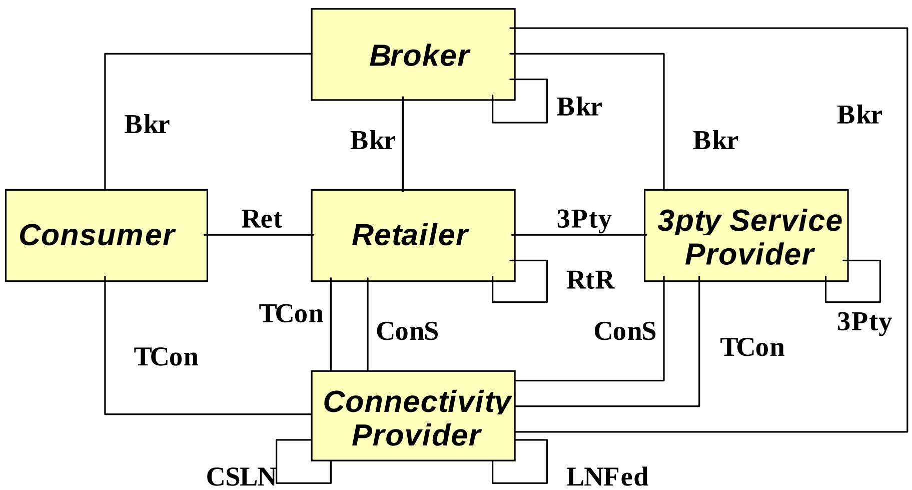


Figure 2.2: TINA Business Model

The Retailer can either provide the TINA service autonomously or can make use of 3rd Party Service Provider(s) to offer a service. The latter can be regarded as a service "wholesaler", which does not deal directly with Consumers. As an example, a mobile phone service provider would provide the information about football match results, but this content is actually dynamically sourced from a 3rd party content provider as needed. The mission of the *Broker* business role is to provide stakeholders with the information they need to find other stakeholders in a TINA system. In contrast to the Retailer, the Broker should be considered as a directory service provider rather than a one-stop shop for TINA services.

These four business roles have been explicitly separated from that of the *Connectivity Provider* stakeholder, whose role is to manage a transport network and to offer a technology-independent connectivity service to the other business roles.

Figure 2.2 also identifies the relationships (or reference points) between the different stakeholders. The Broker (Bkr) reference point defines the relationship between a Broker and several other stakeholders, namely Consumer, Retailer, 3rd Party Service Provider and Connectivity Provider. The Retailer reference point (Ret) identifies the relationship between consumer and retailer. 3rd Party Service Provider Reference Point identifies the relationship amongst 3rd Party Service Providers as well as between Retailer and 3rd Part Service Provider. Connectivity Service Reference Point (ConS) and Terminal Connection Reference Point (TCon) identifies the relationships between Connectivity Providers (who provides transport services between stakeholders) and Consumer, Retailer and 3rd Party Service Providers. Layered Network Federation (LNFed) and the Client Server Layer Network (CSLN) reference points describe the relationships amongst Connectivity Providers in supporting cooperative connectivity across providers.

2.4.1.2 TINA Service Architecture

A TINA system consists of application software components, which are deployed on the Distributed Processing Environment (DPE) [Chapman 1995]. The DPE provides a software layer on top of the native computing and communication environment, which hides details of the underlying (typically heterogeneous) technology and distribution concerns from the application components. In this way, it supports the construction of portable, interoperable code.

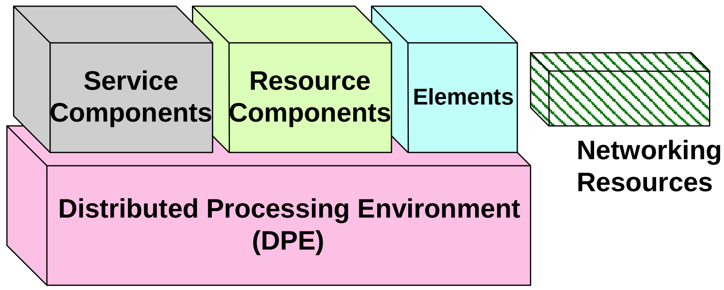


Figure 2.3: TINA components

In order to achieve a good structure, modularity and software reusability, the TINA application components are divided into three categories, as shown in Figure 2.3. *Service components* address the core functionality of TINA services, including access and management capabilities. These components are deployed in the domains pertaining to Consumer, Retailer, Third Party Service Provider and Broker stakeholders. The TINA Service Architecture also details the modelling concepts used to define and design service components. Service components that require a connectivity service can use facilities provided by *resource components*, which offer high-level technology-independent abstractions of the underlying transport network in order to utilise and manage the network's resources. These components are deployed within the Connectivity Provider stakeholders' administrative domains. The TINA Network Resource Architecture (NRA) provides a set of generic concepts to describe transport networks in a technology-independent manner. The TINA NRA is heavily influenced by Telecommunication Management Network (TMN) standards. *Element components* are software representations of individual switching and transmission resources, such as switch fabrics and transmission equipment. The identification and definition of individual element components is considered to be outside the scope of TINA. An in-depth description of TINA's architecture can be found in [Wade 1999b], [Pavon 1996] and [TINAC].

2.4.1.3 TINA Service Architecture's Contribution and Influence

The TINA industrial consortia (1993-2000) was not as long lived as that of DMTF and the TeleManagement Forum. However TINA promoted a number of issues which have been subsequently taken up and progressed by the others, namely: (i) a critical analysis of the business/enterprise stakeholders, their relationships (reference points) and their influences on the resultant management architectures; (ii) its expressed objective towards component oriented architectures; and (iii) the use of mainstream distributed middleware services to support management system and component communication.

TINA's business modelling work has influenced the stakeholder representation in the standards fora e.g. TeleManagement Forum eTOM. TINA's work in adopting mainstream distributed object technology, has been take-up by many management system developers and vendors. However the middleware churn has now progressed toward EJB, .NET and Web Service based solutions. The TINA service architecture and concepts can now be found in many products in the industrial market.

2.4.2 TeleManagement Forum Service Architecture Initiatives

The TeleManagement Forum is a not-for-profit global organization that provides strategic guidance and practical solutions to improve the management and operation of communications services [TeleManagement Forum]. Its membership comprises incumbent and new-entrant service providers, computing and network equipment suppliers, software solution suppliers and customers of communications services. The TeleManagement Forum has had many initiatives in the telecommunication service domain. One of the most influential with regard to developing management services and components has been their Enhanced Telecommunication Operations Map (now called eTOM) which has been progressed to *de-jure* standards as ITU-T M3050. A second related initiative from TeleManagement Forum is the 'Next Generation Operational Support System (NGOSS) which attempts to define architectural aspects for development of OSS[1] systems. Finally, the TeleManagement Forum has recently investigated the NGOSS life cycle and methodology.

2.4.2.1 Enhanced Telecommunication Operation Maps (eTOM) and Processes

The TeleManagement Forum developed a Business Process Model (BPM), which attempts to map out the high-level telecom business processes. The eTOM is presented as a hierarchical (top down) approach to modelling business processes (Figure 2.4). At the top level (Level 0 Processes), eTOM identifies three vertical process areas: (i) Strategy, Infrastructure & Product; (ii) Operations; and (iii) Enterprise Management. eTOM also identifies four horizontal process areas namely: (i) Market(ing), Product & Customer Processes: (ii) Service Processes involved in developing and managing services; (iii) Resource Processes for managing network and IT resources; and (iv) Supplier/Partner Processes for managing interaction with suppliers & partners. Figure 2.4 illustrates these vertical and horizontal process areas.

[1] An Operational Support System (OSS) is the telecommunications industry term for a management platform and the set of management applications which execute on top of that platform, and which manage a network, subnetwork, or set of software applications/computing environments.

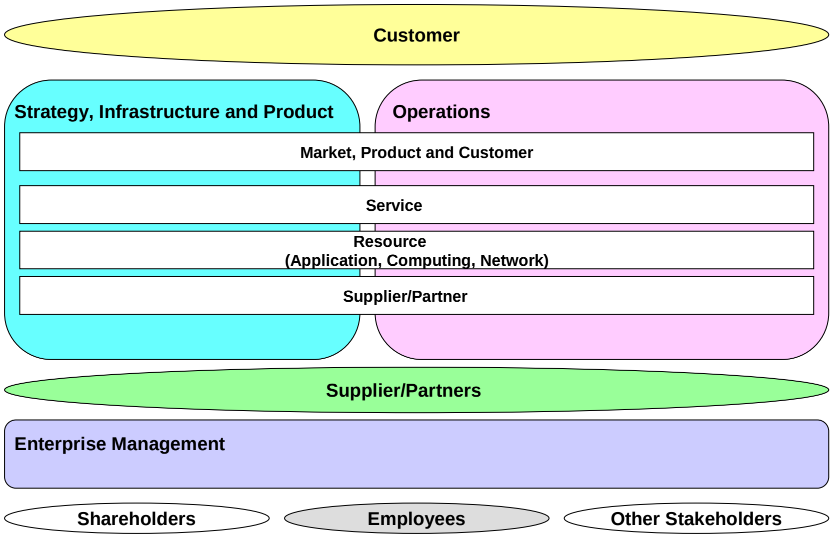


Figure 2.4 eTOM business processes (Level 0)

eTOM decomposes the processes in each of these areas. The Strategy, Infrastructure & Product process are decomposed into vertical process such as Strategy & Commit, Infrastructure Lifecycle Management, Product Lifecycle Management. These vertical processes are divided horizontally into processes relating to Marketing & Offer Management, Service Development & Management, Resource Development & Management and Supply Chain Development & Management. These decompositions are illustrated in Figure 2.5

The Operations Process area is decomposed into four vertical processes namely Operations Support and Readiness, Fulfilment, Assurance and Billing. These vertical processes are divided horizontally into Customer Relationship Management, Service Management and Operations, Resource Management & Operations and Supplier/Partner Relationship Management. Most of TeleManagement Forum’s work has focused on the Operations Processes and the Level 1 processes in this area are further (hierarchically) decomposes into finer grained processes Level 2.

The final level 0 process area is Enterprise Management and is divided into processes related to Strategic & Enterprise Planning, Enterprise Risk Management, Enterprise

Effectiveness management, Knowledge & Research Management, Financial & Asset Management, Stakeholder & External Relations Management and Human Resources Management.

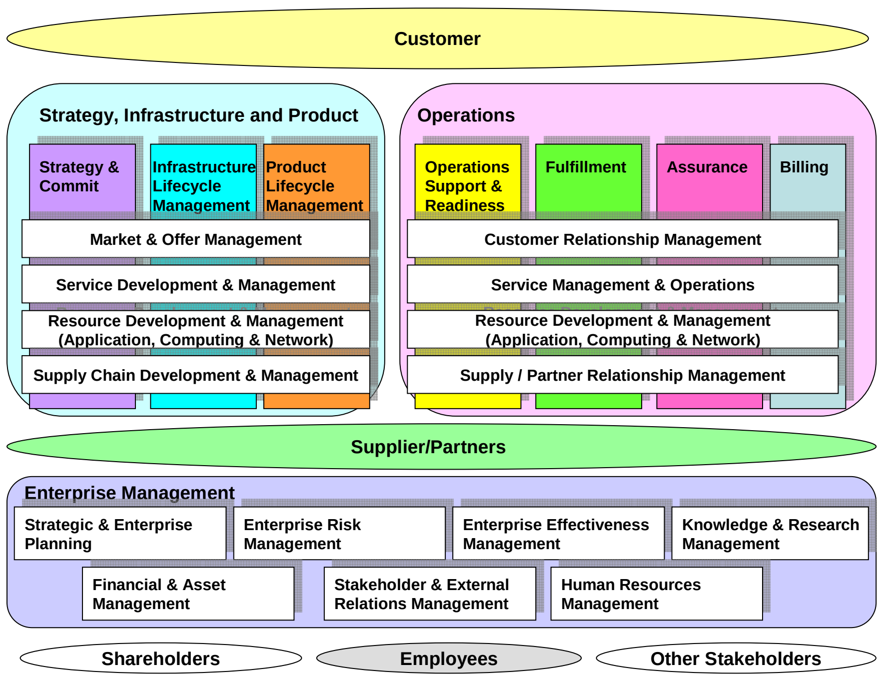


Figure 2.5 eTOM Business Process Framework depicting Level 1 Processes [TeleManagement Forum eTOM 2005]

It is important to note that eTOM is not intended to be prescriptive as to how tasks (within processes) are carried out, how a service provider is organized or how activities and processes elements are sequenced to achieve/implement end-to-end business process flow. It is intended as a guiding reference for service providers in designing or decomposing business processes.

The TeleManagement Forum believes these Flow-Through (vertical) processes to be intrinsic for a Service Provider (SP) to provide services to customers [Wade 2000].

The intention of TeleManagement Forum business process frameworks is to provide a number of cost benefits to the service providers, namely to:

- facilitate better integrated business process interactions between the Service Provider and their customers; also with other Service Providers and network technology providers;

- allow build/buy decisions to be made whereby the SP can perhaps choose to procure Operational Support Systems (OSS) components which support appropriate parts of the business process framework and associated information flows; and

- provide a ready means to drive process automation within the SP environment;

The eTOM Process Modelling Approach

The eTOM provides template based descriptions of the process areas providing a textual description of the objective of the process and a unique ID for each the process. Most of the eTOM processes are defined at level 1 and 2. eTOM was developed using a commercial Process Development tool called CaseWise Corporate Model Builder [Casewise 2006]. For the level 1 and level 2 processes, the decomposition is described in terms of textual descriptions and a hierarchical diagram illustrating the decomposition as depicted in Figure 2.6.

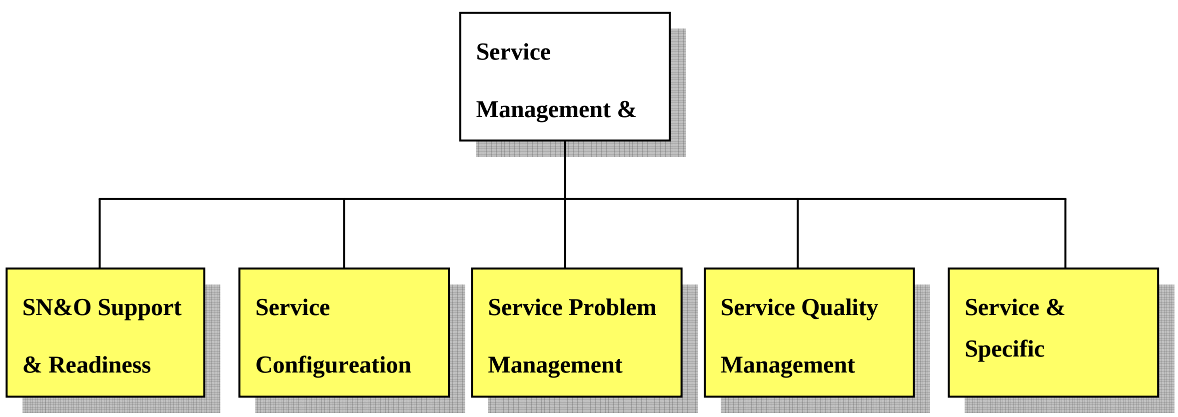


Figure 2.6 Example Representation of Process Decomposition in eTOM (extract from TeleManagement Forum eTOM Annex D 2005)

For detailed description of the process flows eTOM uses flow diagrams represented by Casewise.

2.4.2.2 TeleManagement Forum's Next Generation Operational Support System

In order to provide a more systems oriented view of telecommunication management systems, TeleManagement Forum began work in 2001 on an architectural definition of Next Generation Operational Support Systems[1] (NGOSS). This work (which has been matured more recently in 2006) identifies architectural elements which TeleManagement Forum believes are key to designing technology neutral management systems. Some of the key architectural concepts of NGOSS are summarised in Table 2.1 below. A fuller description of the architectural elements can be found in NGOSS Technology Neutral Architecture [TeleManagement Forum NGOSS 2005].

[1] Operational Support Systems is the conventional name used by telecommunications providers for telecommunications management systems

Architectural Concept	Description
NGOSS Contract	It provides a specification of the operations and behaviour. It exposes the functionality contained in a NGOSS Component. A Contract is structured into four parts: (1) Functional Part: describes the capabilities provided; (2) Management Part: describes the management requirements needed to operate the functional capabilities; (3) Non-Functional Part: defines non-functional aspects needed to provide proper operation of the capabilities (e.g. security, costs etc); (4) Model Part: contains various types of UML models which describe the functional and non functional aspects of the Contract.
NGOSS Component	A NGOSS Component is a software entity that is independently deployable, and that is built conforming to a component software model. It uses Contract(s) to expose its functionality. A component should contain at least two types of contract, namely a contract identifying non-management functions (which describe the main operations offered by the component) and a contract identifying management functions (used to manage the component itself). It can be thought of as a unit of packaging of function(s). It can offer one or more services
Service	Consists of one or more NGOSS Extensible Elements. It can be created, deployed, managed, and torn down by one or more NGOSS Contract (Operations).

Table 2.1 Key NGOSS Architectural Concepts

NGOSS also defines ‘framework services’ which support the distribution transparency of the components in an NGOSS system, as well as supporting sequencing, security, policy etc. This is called the NGOSS Technology Neutral Architecture Specification and is depicted in Figure 2.7. These framework services are not defined in any detail but are depicted in the NGOSS architecture to illustrate how the NGOSS components could be supported.

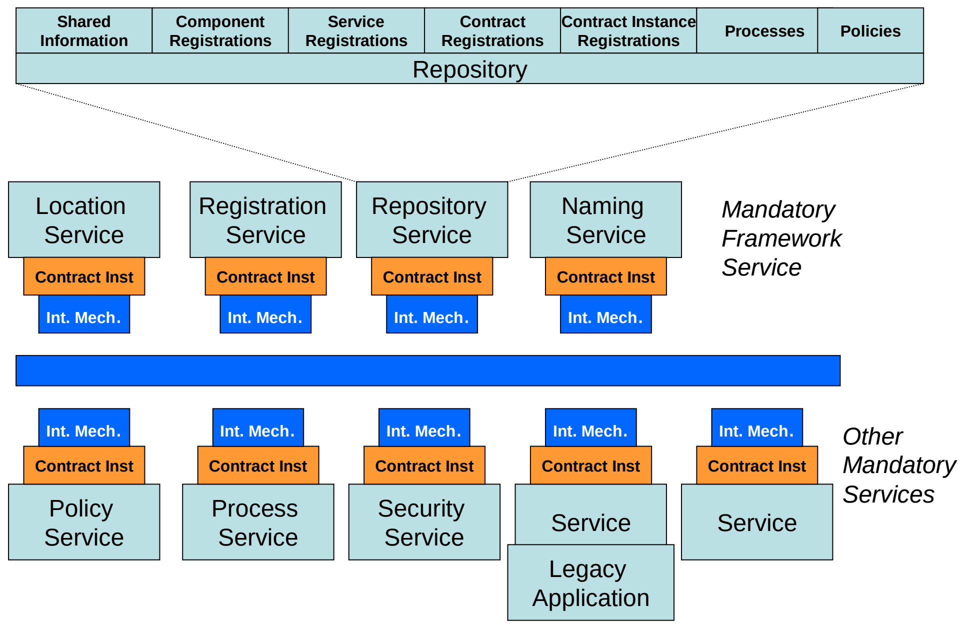


Figure 2.7 NGOSS Technology Neutral Architecture

2.4.2.3 TeleManagement Forum's Shared Information Data (SID) Model

The Shared Information Data (SID) Model provides the NGOSS information model that is a representation of business concepts, their characteristics and relationships, described in an implementation independent manner. The SID Model is intended as a common information model and can be used across telecommunication management applications. Thus the SID Model represents a single (homogeneous) model (or expects other non-Telemanagement Forum information models to be mapped to the SID's Models specification language). The SID Model is used to represent different information perspectives e.g. Business Level Information, System level Information, etc. The SID Model provides an information/data reference model and a common information/data vocabulary from a business as well as a systems perspective and uses UML to formalize the expression of the needs of a particular view. Although the intention is that the SID

Model will fully support all four views much of the work is still being developed and while the documents available today cover a substantial part of a Service Provider's core information needs, they do not yet cover all of them [TeleManagement Forum SID 2005]. Figure 2.8 identifies some of the principle business entities in the SID Model (level 1).

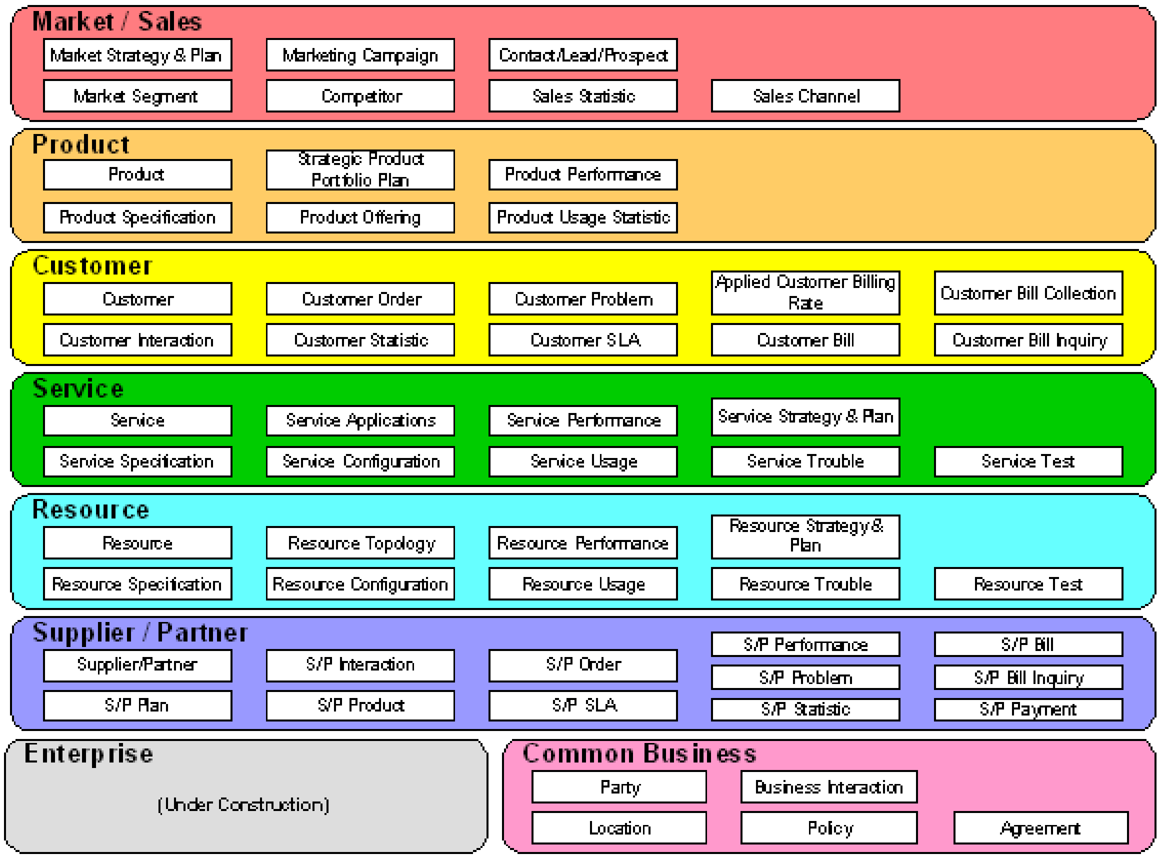


Figure 2.8 SID Level 1 of SID Business Entity Framework

The SID Model has four viewpoints; Business, System, Implementation and Technology. The SID System View is intended primarily for architects, designers and implementers. As has been explained, the NGOSS Business View makes use of the eTOM and the SID to focus on the concerns of the business: goals, processes, entities and interactions. Used together they identify business processes and the information entities needed to support those business processes in achieving the business objectives expressed in the Use Case. In the NGOSS System View, the SID, the eTOM and the NGOSS Architecture are used to focus on the system concerns: objects, behaviour and computational interactions. Here, the SID Model is used to add detail to the artefacts identified in the Business view, as well as to define new artefacts to support the needs of this view. This enables the business processes to be further refined. It also means that the contractual interfaces that represent the various business process boundaries can be identified and modeled. These can be collectively used to define the inputs needed for the implementation view.

2.4.2.4 Contribution and Influences of TeleManagement Forum in Telecommunications Service Development

The TeleManagement Forum's work on eTOM is well accepted by industry as a starting point for the development of management systems. The TeleManagement Forum are currently evolving this work within the enhanced telecommunications operations map (eTOM) which reflects the importance and growth of Internet-style service delivery and Business-to-Business co-operation. The TeleManagement Forum Next Generation Operational Support Systems initiative has been a more recent initiative and attempts to provide designs and models which are implementation technology neutral. Their designs adopt a Unified Modelling Language (UML) based approach to architecture specification. The SID provides the 'information language' which can be used in both the eTOM processes as well as within system level views of NGOSS. This interdependence between SID, eTOM and the design of telecommunication management applications is represented in Figure 2.9 [Strassner 2005].

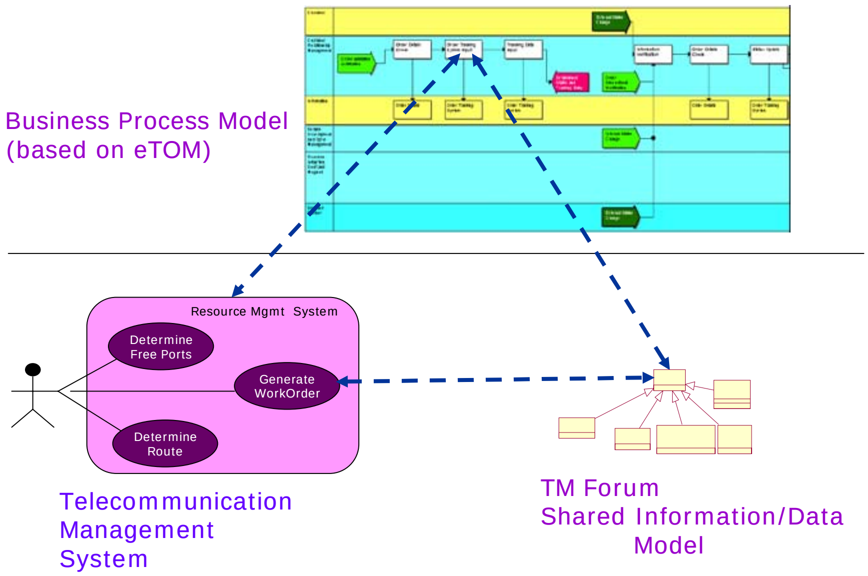


Figure 2.9 Integration Correspondence between eTOM, SID and Telecommunication Management System under construction [Strassner 2005]

The TeleManagement Forum work has encouraged the use of ‘business processes’ as a driver for the development of telecommunication applications. In particular it has modelled processes in the area of Fulfilment, Assurance and Billing. However, there are some important aspects which eTOM does not address. eTOM is a high level process framework which can inform the development of actual business process design and implementation. However, the level of abstraction of eTOM is very high. This is deliberate from a standardization point of view as further decomposition of them would create very complex process descriptions and would restrict the ability of commercial TeleManagement Forum members deciding how they wish to design/implement such processes. Thus eTOM does not provide a set of processes ready for implementation, but rather a starting point for development of management processes. A second limitation of eTOM is that it does not provide a methodology as to how to develop and further refine their process models. The design activities and development processes needed to both refine the models and integrate with existing or new telecommunications management applications is not within the scope of the eTOM. eTOM is focused on defining the telecommunication management business concepts and the business processes themselves, rather than the techniques and development activities which assist design and implementation.

Finally, the TeleManagement models are quite homogenous, meaning that they can be difficult to use within heterogeneous environments, i.e. environments where other standards, not conformant to TeleManagement Standards, are also used. The TeleManagement Forum does have various 'interoperabilty' whitepapers or memorandum of understanding (e.g. with DMTF) but for example, their information models are fundamentally different.

2.4.3 Distributed Management Task Force - Web based Enterprise Management

Founded in 1996, the Distributed Management Task Force (DMTF) is the industry organization that is attempting to lead the development, adoption and unification of management standards and initiatives for desktop, enterprise and Internet environments [DMTF 2004]. Although originally focused on desktop (computing) management, the DMTF has expanded its original remit to the management of the organisations, desktops machines, application management, server management, host management, network management, and telecommunication management. Composed of (predominantly) technology vendors and affiliated standards groups, it attempts to "…enable a less crisis-driven approach to management through interoperable management solutions". More specifically its Web Based Enterprise Management (WBEM) is a set of management and Internet standard technologies developed to unify the management of enterprise computing environments.

WBEM principally consists of three related standards namely: the Common Information Model (CIM), Web Based Enterprise Management (WBEM) and an XML binding for CIM. The first is Common Information Model (CIM), which represents core management concepts, and from which are derived several more domain specific models e.g. systems, devices, applications, networks etc. This model is specified in Model Object Format (MOF) but is increasingly being represented in UML.

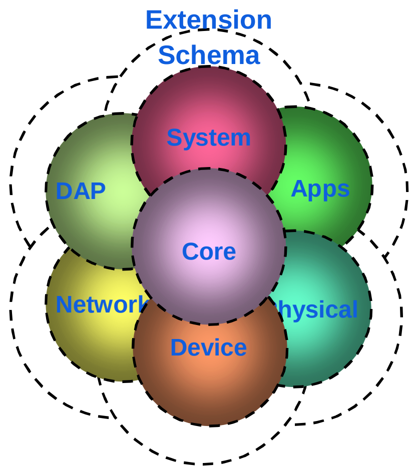


Figure 2.10 DMTF's Common Information Model [DMTF 2004]

The concept behind the Common Information Model (CIM) is to have collections of useful management information specified to model different management enabled artefacts. Thus CIM is a set of management information class specifications, incorporating the modelling of applications, physical aspects, devices, networks, systems and Directory information [Westerenin 2001]. Figure 2.10 depicts the key element of the DMTF CIM schema: the CIM Core Schema, which defines what it is to be a managed element and some of the fundamental elements of managed systems, and the main extensions to this core to allow the modelling of systems, applications, physical networks, devices, networks and Directory Access Protocol. Each of these discrete models inherits from the core model, which contains specifications of management concepts useful to all the models. The Information classes are specified in a syntax called Model Object Format (MOF) which is independent of any particular programming language i.e. can be represented in JAVA or C++ or other any object oriented programming language.

However, in order to manipulate this management information, an access protocol is required. Thus, the second element of WBEM is an XML binding for CIM, so that messages with content based on the CIM model can be passed in XML documents. This allows for relatively easy parsing and generation of messages as XML technology usage becomes ubiquitous.

The third part of the DMTF standards is their usage of HTTP to transport the CIM/XML. HTTP provides a highly flexible management transport protocol for exchanging CIM based, XML encoded management information. The relationship between the three standards is depicted in Figure 2.11

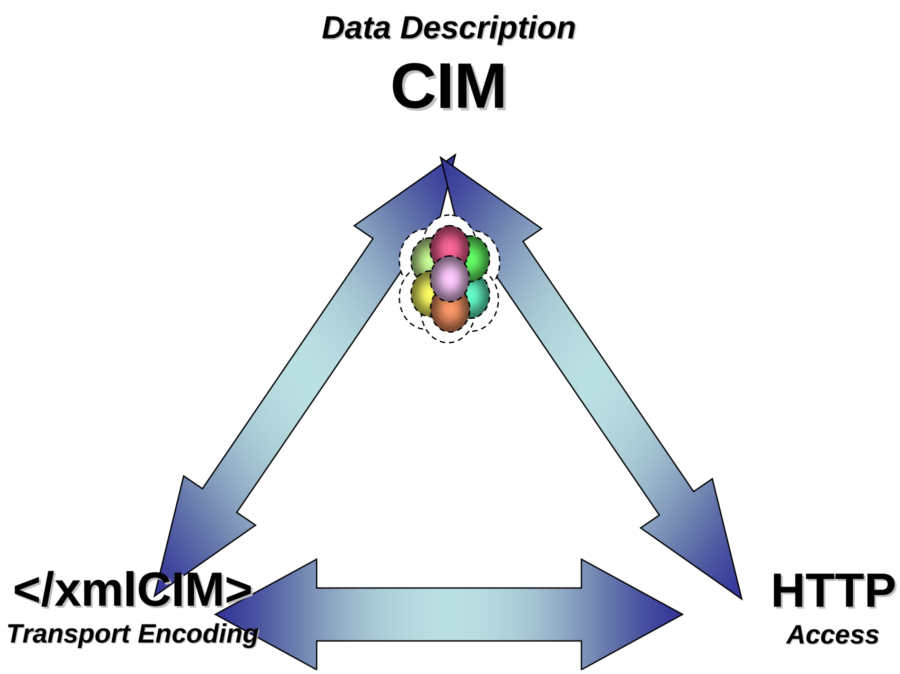


Figure 2.11 Key DMTF Specifications for use of CIM [Westerenin 2001]

2.4.3.1 DMTF Contribution to and Influences on Development Activities for Telecommunications

Although more recent than the other standards/industry fora, the DMTF has seen rapid acceptance of its Common Information Model, in particular by key industry actors such as Microsoft and Cisco. The flexibility offered for the transportation of management information over XML and HTTP, and the wide availability of such technologies, has been a key factor in the inexpensive development of management infrastructure. It was the first of the telecommunication management standards to embrace the use of UML to communicate their Common Information Model (although the reference specification of the CIM schema is specified in a formal MOF language). DMTF provides a very rich but complex set of information models, which although useful, are hindered because of a lack of explanation of how they can be used, in what types of application and in what way. Thus, management system developers frequently struggle to identify the appropriate information objects for their applications, or discover that their application involves multiple sub-models and therefore involve complex selection of subsets of the CIM. DMTF does not provide a process framework over their CIM, which could both validate the presence of their CIM objects as well as provide much more appropriate guidance as to the use of those objects. DMTF provides no methodological guidance as to the design of management applications using CIM as this is considered out of the scope of the standard.

2.4.4 Summary of Service Architectural Influences of Service Management System Design

The survey of the relevant telecommunication management bodies has identified several common trends. Firstly, Service Management Architectures are increasingly adopting mainstream distributed technology to provide the basis for management system integration and interoperability. As these mainstream (middleware) technologies have evolved, so have the various telecommunication management (standard) architectures and their implementation technology selections. The second common theme has been the take-up of component oriented management system realization. Most of the standards reflect a component oriented ethos to management system realization. A third common theme is the adoption of UML as a design model representation for both management information and systems. UML is being advocated for its technology implementation neutral aspects as well as its expressive power. A fourth theme has been the need to address the business modelling aspects of the management system to reflect the business context stakeholders, actors, roles and constraints. This is very important as it reflects the management context, and standardization influences which operate within the telecommunication management sector.

An important aspect common to the DMTF, TeleManagement Forum and TINA work is the omission of methodological or development activity guidance as to how to develop such systems. These standards have focused on 'what the system should do' and 'what information the system should use' rather than on 'how to develop a system to do it' and 'how its development can be influenced by the information which it can expect to be able to access'. In general, standardization bodies/consortia (in telecommunications management) avoid trying to standardize or recommend a particular development methodology because this could limit the commercial competitiveness of commercial members of the standardization body or consortia. However, the most recent exception to this is the TeleManagement Forum's very recent work on a Lifecycle Model and Methodology [TeleManagement Forum Lifecycle 2006]. The need for such a methodology for development of TeleManagement conformant systems was evident due to the complexity of the TeleManagement Management Architecture (called NGOSS) and the related design models. As this is a methodology, it is analyzed more closely in the next chapter (Chapter 3) which addresses the development of methodologies for telecommunication management systems.

2.5 Model Driven Architectures

The Model Driven Architecture (MDA) is an initiative of the Object Management Group (OMG) to provide support in the use of various models (and notations) in the development of software [OMG 2003]. It is intended to support portability, interoperability and reusability by separating the specification of the operation of a system from the details of the way that system uses the capabilities of a platform on which it might execute. It is termed *model-driven* as it provides a means for using models to direct design, construction, deployment, operation, maintenance and modification of a system. MDA is designed around the core of models of Unified Modelling Language (UML) , the MetaObject Facility (MOF), XML Metadata Interchange (XMI), and the Common Warehouse Metamodel (CWM) [OMG 2003]. These modelling languages are intended to be used to capture the various system models and to support the transformation of models into executable code. Figure 2.12 illustrates the application areas to which MDA is intended to be applied.

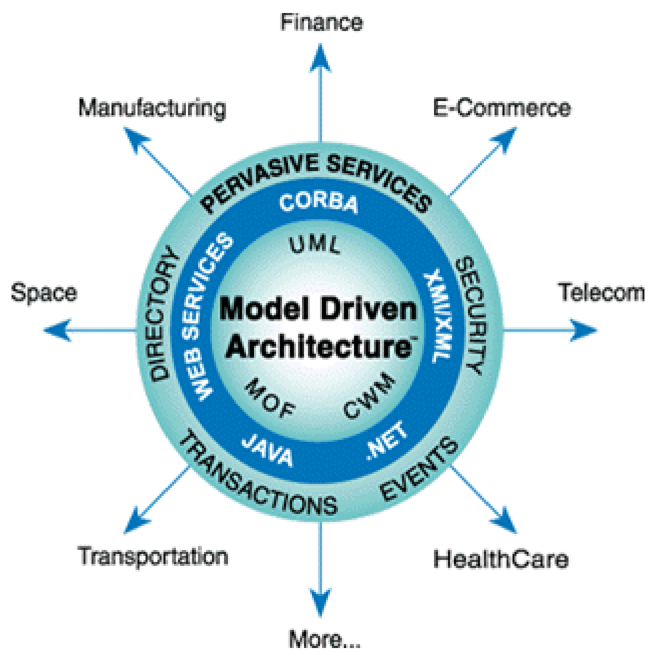


Figure 2.12 Model Driven Architecture [OMG]

A key part of the Model Driven Architecture approach consists of the transformation of Platform Independent Models (PIM) to Platform Specific Models (PSM) and programming code [Siegel 2005]. First the modelling of the application which specifies every detail of its business functionality. This model is called the Platform Independent Model (PIM) and is expressed in UML. This model of the system exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type. The second model is called the Platform Specific Model (PSM) and is a model representation of the system combining the specifications in the PIM with the details that specify how the system uses a particular type of platform (which is intended to support the system). The transformations from PIM to PSM typically involve marking up the PIM and using various MDA tools to generate the PSM. Figure 2.13 depicts the different models and their transformation sequence.

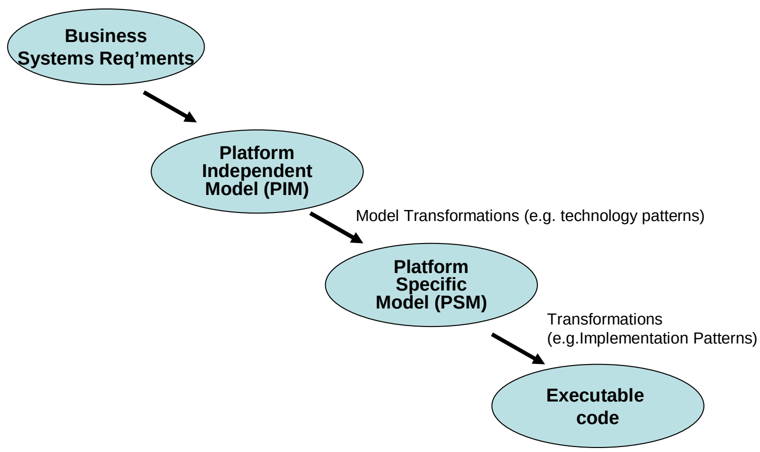


Figure 2.13 MDA Model Transformations

MDA is still maturing and there are many problems and challenges with implementing the approach. For example transforming the PIM to the PSM is non trivial and frequently requires "the poor developer, misled by the – *all you need is UML* – hype, is stuck having to debug and develop code that a tool generated" [Thomas 2004]. For these reasons many leading commentators believe that MDA has much potential but it could be as much as ten years before the original goal is achieved [Bezivin 2003].

2.6 Service Oriented Architectures

The IT industry (and latterly the Telecommunications Management Industry) has begun to move toward 'Service Oriented Architectures'. Service Oriented Architectures can be viewed from multiple perspectives. Twardes suggests that, from a business perspective it is "a set of services that a business wants to expose to their customers and partners, or other portions of the organization" [Twardes 2006]. Whereas from an architectural perspective it is "an architectural style which requires a service provider, requestor and a service description. It is a set of architectural principles, patterns and criteria which address characteristics such as modularity, encapsulation, loose coupling, separation of concerns, reuse, composability and single implementation".

Sayed Hashimi [Hashimi 2003] provides a more exacting definition of a service as "an exposed piece of functionality with three properties:

1. The interface contract to the service is platform-independent.
2. The service can be dynamically located and invoked.
3. The service is self-contained. That is, the service maintains its own state".

However, common key aspects of a Service in a Service Oriented Architecture are that: (a) the service represents how it wishes its consumers to use it; (b) the service has location transparency, i.e. it is loosely coupled (independent of the execution platform); and (c) the service is self-contained. One of the key reasons for its rapid acceptance is the definition and impact of Web Service Architecture [W3C 2004]. A Web Service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format, specifically WSDL [WSDL 2001]. Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Because it has been adopted by the vast majority of major software vendors and corporations, its value has already outstriped any previous attempt at inter-enterprise collaboration (including electronic data interchange) [Jones 2005]. In fact it sometimes needs to be re-emphasised that the Web Service Architecture is only one instance of a Service Oriented Architecture, although currently it is by far the most significant example of a service oriented architecture.

However some have criticised or expressed concerns that Service Oriented Architectures are not the 'panacea' for Telecommunications Management architectures [Strassner 2006, Caruso 2006]. In fact Caruso suggests such architectures may not suit: (i) critical real-time, high performance interfaces e.g. Call processing interface in an intelligent network element; (ii) Homogeneous standardized environments; (iii) Static environments with little or no change; (iv) Existing interfaces that will not provide business value to the customer or lacks customer demand; (v) Bulk data synchronization processes; or (iv) Backup and restore processes.

The TeleManagement Forum also identifies that in fact Service Oriented Architectures, are a specialization of Distributed Interface Oriented Architecture (DIOA), as the service definition just specialized the kind of information contained in an interface [TeleManagementForum NGOSS 2005]. Their claim is that in fact their DIOA architecture is more general than a Service Oriented Architecture but can be instantiated as a Service Oriented Architecture.

2.7 Approaches and Technologies for Integrating Telecommunication Management Systems

There are several widely used distributed component implementation technologies, possibly the most established of these standards being CORBA [OMG CORBA] and Enterprise Java Beans [Sun EJB]. Other developments have seen telecommunication providers experiment with .NET based solutions and Web Service based solutions. Similarly there are many approaches to integrating such components [Wade 1999a] e.g. Business Object, Event Driven Integration and Workflow. This section briefly highlights Workflow as a Component integration technology. This represents one of the most commonly used integration technologies in use in telecommunications management systems. This technology is also used in the implementation of integration of the components, which were designed using the methodology proposed in this thesis.

2.7.1 Workflow Based Component Integration

There is a growing awareness that workflow management tools and techniques could provide a vital element in the co-ordination of distributed components within different provider domains while allowing greater flexibility and the necessary degree of autonomy [Wade 2000]. Because of the introduction of new services, new relationships with other service providers or new functionality or equipment, the management components from service provider and network provider systems must be capable of adapting rapidly to changes in the way the business process is executed.

A workflow management system is a system that defines, manages and executes workflow processes through the execution of software whose order of execution is driven by a computer representation of the workflow process logic [WfmC2001]. Workflow technology incorporates the benefits of co-operative information systems, computer-supported co-operative work, GroupWare systems, and active databases. Workflow management technology addresses the following requirements:

- Improved efficiency, leading to lower costs or higher workload capacity;
- Improved control, resulting from standardization of processes;
- Improved ability to manage processes; identification and analysis of problems ;
- Cost reductions, where cost can be a euphemism for staff;
- Increased quality or capacity while controlling costs;
- Construction of unique customised business processes to deal with specialised management work practices;
- Improved information distribution, and elimination of the delays caused by the need to move hard copy information around the organization; and
- Reduced bureaucracy, improved quality of work, decreased cycle times, and acquisition of better management information about business processes.

Thus workflow management can be considered an attractive technology for integration and interrelation of telecommunication management components. The purpose of applying workflow management technologies in the service management problem domain is to integrate and re-purpose management components that resolve telecommunication business problems, and to automate telecommunication management services. Such enhancements, i.e. the interrelation of service management components, reduce the business process complexity, improve resilience and improve the overall performance of the network providers’ business process.

There are many differences between the architectures, which are used by workflow systems. However, most of the workflow systems fall into one of two broad categories:

- Forms and messages based workflow systems which performs electronic routing of forms to user's e-mail in-boxes; and
- Engine based workflow systems, which communicate with humans or components via specialised client software.

2.7.1.1 Workflow Engines

A Workflow Management System (WFMS), as defined by the Workflow Management Coalition, is "a system that defines, creates, and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke applications (or components)" [WfMC 2001]. A workflow engine is the basic workflow management control software. It is often distributed across a number of computer platforms to cope with processes, which operate over a wide geographical area.

The workflow engine controls the flow of work (sequences of management activities, which form a management business process) through the system by interpreting the management process rules to determine the scheduling of required activities, and invoking the relevant management components. The engine is responsible for:

- Business process creation, deletion, and management of process execution from instantiation through completion;
- Control of the activity scheduling within an operational (business) process;
- Interaction with management components and/or human resources (which execute the required management activities); and
- Monitoring and control of the management processes in execution.

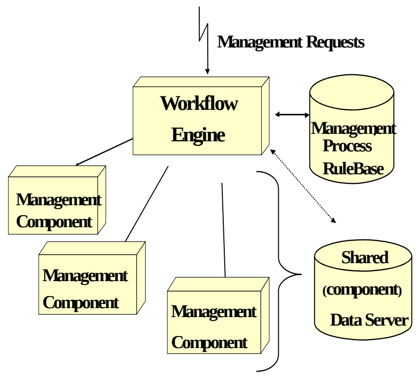


Figure. 2.14. General Workflow Architecture

Figure 2.14 illustrates a generalised workflow engine, which accepts a (management service) request and based on its process rulebase, invokes the correct sequence of components and stores application specific values in a shared data server.

Many of the workflow products simply provide a means of graphically representing a business process using techniques such as dataflow, digraph, flowchart, network, orgcharts, pertcharts etc. Others are data management systems, which use e-mail, imaging, databases, electronic forms, engineering drawings etc. to collaboratively process documents or data. Groupware also forms part of this group, with IBM Lotus Notes being a good example.

All of these systems have an emphasis on office processes, e.g. imaging, document routing, enhanced mail. However a number of limitations are evident with these types of workflow systems [Sheth 1997], [Sheth 1999].

(i) Lack of support for heterogeneous computer systems;

(ii) Incompatibility between workflow products;

(iii) Failure to capture distributed/true nature of infrastructure in business model;

(iv) Scalability not achieved;

(v) Very little support given towards fault-tolerance and reliability.

2.7.1.2 Workflow Standardization, Languages and Representation of Business Processes

The standardization of workflow systems has been on-going since 1993 with the formation of the Workflow Management Coalition WfMC (an industrial consortium that set about standardising an architecture for workflow engine based systems, and several interfaces for application invocation, process definition, process management and system interoperability [WfMC2001]. In 1998, the OMG ratified the definition of a workflow facility, which was based on the WfMC standards. However, as workflow is of technological interest to many different communities including the workflow document community, the business process modelling community, and the software engineering community, agreement and full standardization has been very difficult.

The result has been several standards and evolutions, which are typically influenced by underlying technology trends. These include the XML oriented Business Process Language (BPL) and Business Process Execution Language BPEL (IBM). More recently, this standard has been developed to define a notation for specifying business process behavior based on Web Services, called WS BPEL [Oasis 2005]. Also recently, the Object Management Group (OMG) has taken on the standardization of a Business Process Modelling Notation (BPMN) as part of its integration of Business Processing Languages within its Model Driven Architecture. Both of these languages are like scripting languages. Also the set of primitives available for sequencing activities (within a workflow defined by these languages) tend to have similar basic elements, and tend to differ in the specification language notation and in the visual representation. Both BPEL and Web Service Architectures are now being combined to provide business process execution above reusable enterprise services [Pasley 2005].

In parallel to this, the W3C is also defining Web Service Choreography standards which focus on the coordinated interactions between agents. This choreography specifies various web services interaction needed to execute a desired overall application or behaviour. W3C has just recently published its ‘candidate recommendation’ for a Web Services Choreography Description Language (WS-CDL). This is an XML-based language that describes peer-to-peer collaborations of participants. It does this by defining, from a global viewpoint, their (the participants) common and complementary observable behavior. The idea of the choreography is that the ordered message exchanges (between participants) result in accomplishing some desired business goal or process in a distributed fashion. [W3C 2005].

2.8 Summary

This chapter has presented an analysis of the key stakeholders and actors in the value chain for telecommunications service management. The chapter identified the key drivers and influences which shape the way telecommunication management systems are developed. The chapter also surveyed the principal standardization bodies which scope and define key management functionality and architecture. The chapter identified different aspects of each standard which have high potential for influencing the architecture and design of telecommunications management systems namely:

Business Modelling:	TINA C Enterprise/Business Model Concepts
Information Modelling:	DMTF Common Information Model, TeleManagement Forum Shared Information/Data Model, IETF SNMP MIB II
Architecture:	TINA Service Architecture, TeleManagement Forum NGOSS, DMTF WBEM

The chapter drew from these standards key aspects which need to be considered within a development methodology for designing service management systems and their constituent components. The chapter also identified key architectural influences from mainstream Information Technology industries. The architectural influence of MDA and Service Oriented Architectures are influencing both the telecommunication standards and the development approaches within the telecommunications industry.

Finally the chapter outlined workflow as a key integration technology for telecommunications management, identifying the benefits, requirements and difficulties in current state of the art workflow approaches and standards. The workflow approach has the ability to integrate the execution of business processes with the flexible invocation of component oriented software and services.

The next chapter will focus specifically on development methodologies and design processes which will underpin and inform the development of a methodology for business process driven, component based, and telecommunications management systems.

3 DEVELOPMENT METHODOLOGIES FOR TELECOMMUNICATION SYSTEMS

This chapter presents an analysis of existing development methodologies for telecommunication management. The chapter outlines the major trends and principal features of development methodologies, which have emerged both within the telecommunication community or have been adopted within the community from mainstream Information Technology/Software Engineering. The chapter also identifies key issues in component and system development of these methodologies and concludes by identifying trends and synergies for telecommunication management development.

3.1 Development Methodologies for Telecommunication Management

Choosing a software development methodology for a telecommunication software project can sometimes result in a 'religious' war where developers originate from different computing and telecommunication engineering backgrounds. However, the skills required for telecommunication software and computer software development are increasingly overlapping [Wade 1999]. Traditional telecommunication development methodologies are less appropriate where the telecommunication management software is more independent from the low-level transmission or protocol handling layers e.g. telecommunication service management, but are strong where signalling and simulation (verification) is required. However, it is not always clear which methodologies to choose for each telecommunication management development project or product.

The past fifteen years has seen two recognisable streams in the evolution of methodologies for telecommunication management systems development. The first stream of evolution originates principally from the telecommunication industry itself, and involves such methodologies as the development of management interface for TMN

(M3020), Specification and Design Language (SDL-92), the work of ITU Study Group 10 and the Open Distributed Processing (ODP) standards [ISO 1997]. The second stream of evolution originates from the general computing software industry and includes several generations of Object Oriented Analysis & Design Methodologies, and Unified Modelling Language (UML). The next two sections survey the evolution of development processes within each stream and provide a comparative analysis of the state of the art in methodology.

3.2 Object Oriented Software development methodologies in the general Computing area

In general, software development methodologies focus on subsets of the following: system conceptualisation, system requirements and benefits analysis, project adoption and project scoping, system design, specification of software requirements, architectural design, detailed design, unit development, software integration & testing, system integration & testing, installation at site, site testing and acceptance, training and documentation, implementation, maintenance [CTG 1998]. The late eighties and early nineties saw the rise in usage of many different software design methodologies. In general, the principals among the 'second generation' of these methodologies are Rumbaugh's Object Modelling Technique (OMT), Ivar Jacobson's OO Software Engineering (OOSE) and Grady Booch's Object Oriented Analysis and Design (OOAD) methodology. The late nineties and the beginning of this century have seen the prominence of 'development process frameworks' which attempt to provide customisable development process elements from which new development processes can be constructed. This section first identifies the early trends in software process development and concludes by presenting and comparing two different approaches to 'process frameworks' realisation.

3.2.1 Second Generation Methodologies

Booch's OOAD method defined the notion that a system is analysed as a number of views each represented as different diagrams. The design process in this method included both a macro and micro view of the system under development. The Object Modelling Technique (OMT) is a rather straightforward design process based on a requirements specification. The method describes a system by means of a number of models: the object model, the dynamic model, the functional model and use-case model, which provide complementary views of the system under construction. OOSE and Objectory methods are both based on use cases, which define the initial requirements of the systems as seen by an external actor. These use cases are then implemented in all phases of the development, including system testing where they are used for verification. OOSE follows the now classical development cycle of use case model, analysis, design, implementation and testing. Objectory has also been adapted for business engineering, where it can model business processes. Figure 3.1 illustrates the general software development lifecycle representing the capture of requirements and the specification of requirements document(s) (statements), the analysis of these requirements to provide a basis for the (object) analysis model, the development of the design model, the implementation of software and the testing of this software. The diagram also identifies the derivation relationship or 'trace' from one model to the previous one. However, changes or amendments to any model need also to be reflected back to the requirements statement so that the completed/implemented models are consistent with the requirements statements.

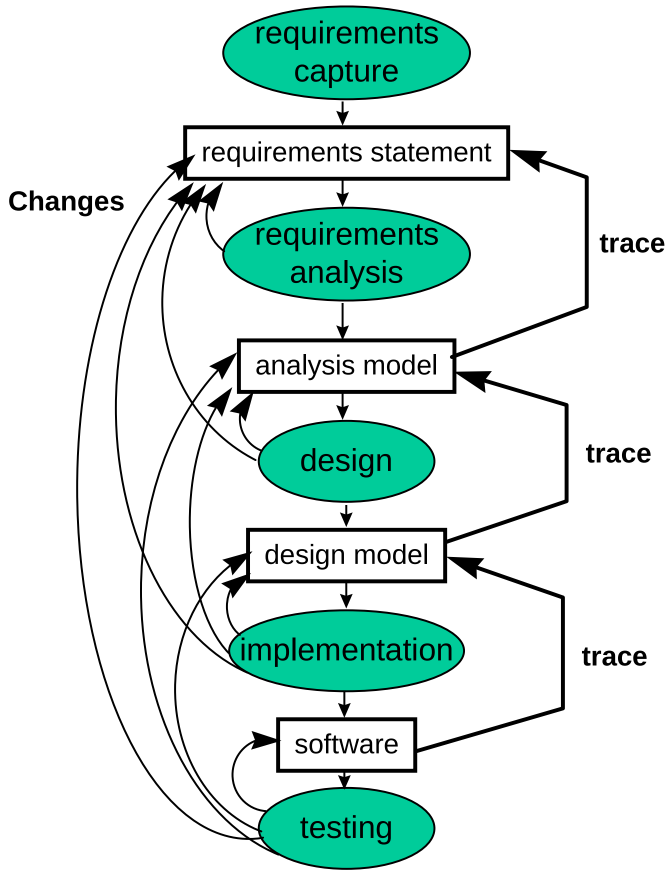


Figure 3.1: General Software Development Process [Wade 1999, Lewis 1999]

A later object oriented methodology was the FUSION methodology developed at HP [Coleman 1993]. This is sometimes referred to as a 'two and a half' generation methodology as it is based on the experiences of previous methods and includes mechanisms for specification of operations and interaction between objects and contains a large number of diagramming techniques.

Each of the methodologies had their own unique (diagrammatic) notation, design process and Computer Aided Software Engineering (CASE) tools that support both these notations and design processes (sometimes). By the late 1990s the dominant trend in object oriented notation and distributed system specification has been to harmonise existing approaches rather than develop brand new modelling techniques. This has been realised in the standardization of a Unified Modelling Language UML. UML is a set of modelling notations, which are independent of any software development process. It specifies the modelling notation and the semantics underlying this notation. UML has utilised and extended modelling elements from Rumbaugh's Object Modelling Technique [Rumbaugh 1991], Jacobson's Object Oriented Software Engineering Methodology [Jacobson 1992] and Booch's Object Oriented Analysis and Design [Booch 1993] methodologies, as well as other lesser-known modelling techniques. The UML notations were originally based on these three principal modelling approaches. The Object Management Group (OMG) has recently accepted UML as a set of standard modelling notations and have progressed their development[1].

The distinction between a development process (method or methodology) and a modelling language is important. The development process is an explicit way of structuring one's thinking and actions. It tells the various actors in the development process what to do, when to do it and why to do it. A modelling language is a set of rules defining (one or more) notations. UML itself is independent of any particular development process. It is unlikely that a 'standardised' development process is possible for all software systems because different application domains exert significantly different requirements, both technical and non-technical. It is envisaged that development processes will be tailored for specific application domains or specific CASE tools [Fowler 1997]. Thus, there is no overall 'best' development process because design processes for different types of systems tend to have different foci and emphasise different characteristics.

3.2.2 Rational Unified Process (RUP)

The Rational Unified Process (RUP) is a software engineering process developed originally by Rational Software [Kruchten 2000] and is now supported by IBM. RUP is itself a specialisation of the Unified Software Development Process (USDP) [Rumbaugh 1999]. RUP is delivered online using Web based technology and consists of more than 1000 hyperlinked pages of text and graphics. It provides a proven disciplined (industrial)

[1] UML version v1.4.2 is the internationally accepted ISO Standard called ISO/IEC 19501(accepted in 2005)

process for assigning tasks and responsibilities within a development organisation to design applications and enterprise systems. RUP aims to capture many of the best practices in software development and then attempts to present them in a tailorable form that is suitable for a wide range of projects.

RUP depicts software development in two dimensions: Phases (Inception, Elaboration, Construction, Transition) and Process Workflows (i.e. development activities), which are conducted within each phase. RUP also identifies three Supporting Workflows, which support the co-ordination of the overall development effort called Change/Configuration Management, Management and Environment). Figure 3.2 represents the four phases, the process workflows and also provides an indication of the level of effort devoted to each process work within each phase.

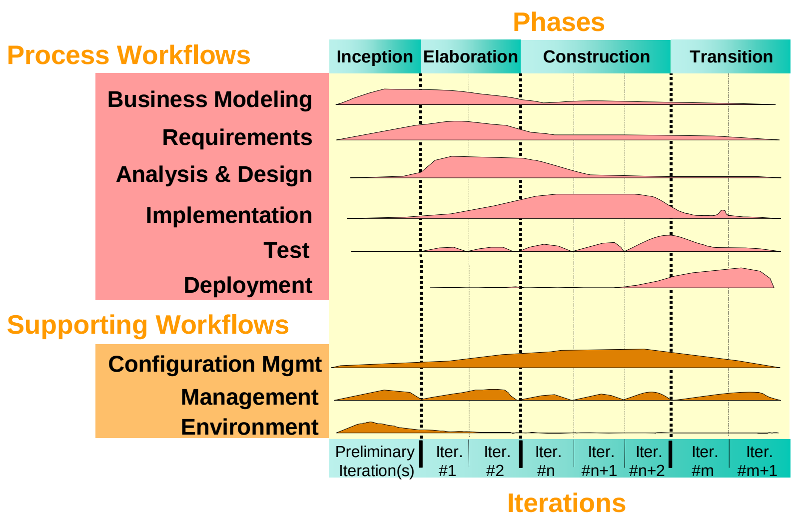


Figure 3.2 Rational Unified Process Lifecycle Model [Kruchten 2000]

RUP is an iterative process in that multiple iterations of the process workflows are expected within each phase. The precise number of such iterations is dependent on the complexity of the solution being developed and the operating context of the development effort e.g. the experience of developers, complexity of application area etc.

The goal of the Inception Phase in RUP is to develop the business case to the extent necessary to justify launching the project [Jacobson 2000]. In this phase, the workflows determine the scope of the system to be developed as well as developing parts of the models that would be necessary to support a proof of concept prototype. However, the business modelling, requirements and analysis workflows are the principle areas of effort in this phase, and the implementation, test and deployment workflows concentrate only on planning activities and infrastructure selection.

The Elaboration Phase has several specific targets, namely the capture of 80% of the required use cases of a system under investigation, commencement of detailed design work, completion of a deployment model for the envisaged system, and completion of about 10% of the implementation work. The Construction Phase should achieve a complete system implementation ready to begin transition to a user community. The Transition Phase involves the deployment of the completed system into its intended user community and the performance of minor fixes and some fine-tuning.

Therefore, the Phases in RUP define 'When' in the software development lifecycle activities should be performed. The 'Who', 'What' and 'How' of the development process are defined within the Process and Support Workflows. Thus in each of the workflows, workers or roles indicating who should carry out development activities are identified, how development activities should be performed and what artefacts (models, design elements) are required to be developed. The Process Workflows defined in RUP are:

(i) Business Modelling,

(ii) Requirement Management,

(iii) Analysis & Design,

(iv) Test,

(v) Deployment;

and the Supporting Processes are:

(i) Change/Configuration Management,

(ii) Project Management and

(iii) Environment.

A more comprehensive overview of important aspects of RUP is contained in [Kruchten 2000] and a broader description of UML based development processes is presented in [Jacobson 99].

3.2.3 Analysis of RUP

RUP is regarded as a Process Framework rather than an actual development process because it specifies many modelling activities and workflows, uses many UML notations and suggests the development of an extremely large number of development artefacts (i.e documents, descriptions, models, etc). This led to criticism of RUP as being overly complicated and difficult to apply to software engineering projects [Hesse 2003] [Wagner 2003] [Graham 2000]. Hesse, in particular suggests its phases and workflows are an unnecessary duplication and cause confusion. He identifies that there is almost a one-to-one mapping between phases and process workflows (see Figure 3.2) e.g. most of requirements workflow is in the elaboration phase, most of the implementation workflow is in the construction phase and most of the deployment workflow is in the transition phase. He concedes that there is spillage of some of these workflows into a subsequent phase but they predominately exist in only one phase [Hesse 2003].

Hesse also points out that RUP should actually be more ‘architecture centric’ and suggests that the workflows (i.e. the design activities) and development iterations should be centered around actual architectural units e.g. components, rather than around the entire system. Others have dispensed parts of the RUP framework and selected a basic profile for software development [Gallerd 2001].

Another criticism of RUP is that it focuses on development of OO software systems, rather than components. Most of the development effort focuses on modelling classes and objects and structures these into folders of class libraries. This approach although useful does not seek to define reusable ‘design time’ components (which we will term building blocks). Rather, components, in RUP, are purely a deployment time instance of classes and objects on particular nodes.

A third aspect of RUP is that although it is intended to be used to develop specific development processes, it does not actually provide guidance as to how such development processes can be refined from the RUP framework [Abrahamsson 2002]. In fact when such specific processes are being (re)defined, there is excessive effort required in determining the resultant artefacts needed during development. For example, suppose a customised RUP process was intended to be more incremental and prototype based (i.e. it need to involve early research with rapid prototyping and subsequent incremental development). It is very difficult to determine which workflows of the design and implementation phases are most appropriate to include and which needs to be deleted. Also, as the RUP activities and artefacts are so inter-twined, removing an activity or design, frequently involves significant redefinition of documentation and related design activities. Also, in the example of making RUP more of an 'incremental[1]' process, it has significant impact the focus on the design work i.e. focusing on the particular separate component designs rather than on the overall system [Hesse 2003].

Finally, RUP does not explicitly define the development of 'business processes'. Although it does define workflows for behaviours, these are typically at the class or method level, and not at the granularity of business processes. The most similar aspect of RUP to business process modelling is in the "use case" modelling and "activity" modelling during requirements analysis/management, where these abstract use case scenarios can describe activities and activity sequences.

3.3 Development Processes in use today

Methodologies or Development Processes tend to differ based on the 'feedback and control methods' employed during development, the timing of activities, the kind of artefact developed by the methodology and the coverage of the lifecycle of software development (i.e. from conceptualisation to maintenance). A survey of process models

---

[1] Incremental development processes or agile processes are defined more clearly later in this chapter. However, in general they are to development processes which focus on particular architecture or component and support rapid development/prototyping of that component [Larman 2003].

was carried out by the Centre for Technology in Government (University of Albany and SUNY) which identified three primary approaches in use today, namely ‘Ad-hoc development’, waterfall model and iterative process [CTG 1998]. ‘Ad-hoc development’ tended to be the way early software systems were developed, and involved chaotic and haphazard development activities which relied heavily on the skills and experience of the individual staff members performing the development. The survey indicated that such methods are still in use today, typically for small projects or for certain subsets of software development.

Waterfall based methodologies typically consist of conceptualisation, analysis, design coding and testing activities. The waterfall model is attributed to providing a theoretical basis for other development methodologies. However its criticisms have focused on the fact that most software development does not follow such a sequential path and that problems only identified at the coding stage, but which require redesign to take place, incur very high cost. This is, however, quite likely when following a purely waterfall model, since at the start of a project there is often a great deal of uncertainty around customer requirements and system objectives. A third criticism is that there is typically a very long lead time before any working system is available for customer testing and acceptance.

The iterative processes tried to address many of these concerns by dividing the development process into many ‘mini waterfalls’ with the feedback being accrued at the end of one or more iterations and this feedback being used to enhance subsequent iterations. Criticisms of the iterative process include the time required by users (customers) to actually be engaged continually within the process, difficulties in communicating the designs between developers, users and system architects. A third criticism has been the potential for requirements to ‘drift’ or ‘expand’ between iterations, giving problems with deadlines being achieved etc.

The iterative methodologies have many variations: Prototyping Model (a variation of this is Rapid Application Development), Exploratory Model (absence of precise requirements), Spiral Model. A full history of iterative and incremental development process is beyond the scope of this thesis, but can be traced back to pre 1970 origins

[Larman 2003]. An associated style of software development which focuses on iterative and evolutionary (incremental) development are called Agile Development Processes (a term coined by a meeting of 17 process experts which originated the 'agile alliance' [AgileAlliance]). Agile methodologies are development process which are lightweight (i.e. provide the minimal amount of design guidance for the development of the desired artefact), iterative (support the incremental development of a system or components) and should be easy to apply. These methodologies have been traditionally associated with software practices as such as extreme programming. However in recent years several authors have argued that lightweight customisations of RUP can be represented as agile development processes [Jacobson 2005] [Ambler 2006].

3.4 Trends in Telecommunication Management Systems Development Standardization

3.4.1 Telecommunication Standards based methodologies

Several bodies (e.g. CCITT/ISO, TeleManagement Forum) have already addressed problems in the areas of service management. The distinction between service management and network management was recognised initially by the Telecommunication Management Network (TMN) standards [ITU-T 2000]. The TMN architecture (M3010) defines conceptual layers addressing different concerns within a provider's operations organisation, i.e. a network element management layer, a network management layer, a service management layer and a business management layer (as shown in Figure 1.1 earlier). The TMN functional architecture makes distinctions between network element functions, mediation functions to non-TMN compliant network element managers, workstation functions presenting information to human providers, and general operations system functions. It also makes distinctions between different reference points that may exist between these different types of functional units and between functional units within and outside of the same organisational domain. These reference points provide the basis for defining interfaces between implementations of the functional units.

Initially it was assumed that these interfaces would be implemented using OSI Management, i.e. the Common Management Information Protocol (CMIP) used to access managed objects defined in a notation called ‘Guidelines for Development of Managed Objects’. However, later revisions to the TMN standards began encompassing other technologies such as Common Object Request Broker Architecture and IDL defined by the Object Management Group [OMG]. The TMN family of standards also includes methodological guidance on the development of management interfaces (M3020). This is based on the definition of management functions that are hierarchically decomposed into Managed Object definitions. Management functions have been defined for both general functions such as event management and log control, but also for network-oriented management functions, e.g. M3100. Though some of these standards can be reused at the service management layer, the ITU or OSI communities have defined few management functions specifically for this layer.

TMN development environments first appeared in the late 1980s and early 1990s and although they aided management system development, they were characterized as very complex and expensive. Later TMN development tools have improved but the skills required to use them is still quite specialised. It is a source of debate as to whether TMN is still a viable, commercial approach to developing architectures for telecommunication management systems because the usage of these standards has noticeably decreased during the period 2000 onwards. However, the TMN standard [ITU-T 2000] is still the ISO/ITU accepted standard for telecommunication network management.

Another very important contribution to software development methodologies for telecommunication systems is SDL. SDL is a Specification and Description Language standardised by ITU (International Telecommunication Union). The language has been evolving since the first recommendation in 1980, 1984, 1988 and 1992 when Object Oriented features were included in the language. Although SDL evolved within telecommunication, it is becoming increasingly popular in other industries as well. Some examples of applications of SDL outside the telecommunication area include satellite communications and control systems. SDL-92 (Z100) is the version of SDL currently supported by most SDL software tool vendors and is quite widely used by large telecommunication providers for low-level network management systems design and implementation.

Another international body that has performed in-depth studies of service management and its development was the Telecommunication Information Network Architecture Consortium (TINA-C). As mentioned in chapter 2, TINA aimed to develop a comprehensive architecture for telecommunication control and management. It based its modelling approach on Open Distributed Processing principles as defined by ITU (X901). It developed detailed models for the integrated control and management of multimedia service and broadband networks based on existing concepts from TMN, IN and ATM. TINA-C developed internal (unpublished, informal) guidelines for modelling its systems. These are based on the ODP modelling concepts, principally the use of the five ODP viewpoints that separate enterprise, computational, informational, engineering and technology concerns. These viewpoints were supplemented with Object Oriented class diagrams, sequence diagrams and simple block diagrams showing computational component structures and their interfaces. Although now no longer maintained, it influenced the definition of the viewpoints in the TeleManagement NGOSS architecture.

3.4.2 TeleManagement Forum Lifecycle Methodology

More recently, the TeleManagement Forum began work on defining guidance on how to use and deploy NGOSS within an organisation. It envisages a fundamentally NGOSS based solution and within this realm, tries to assist service providers and developers in using TeleManagement Forum's Business Processes Framework, Shared Information Model and NGOSS Architecture.

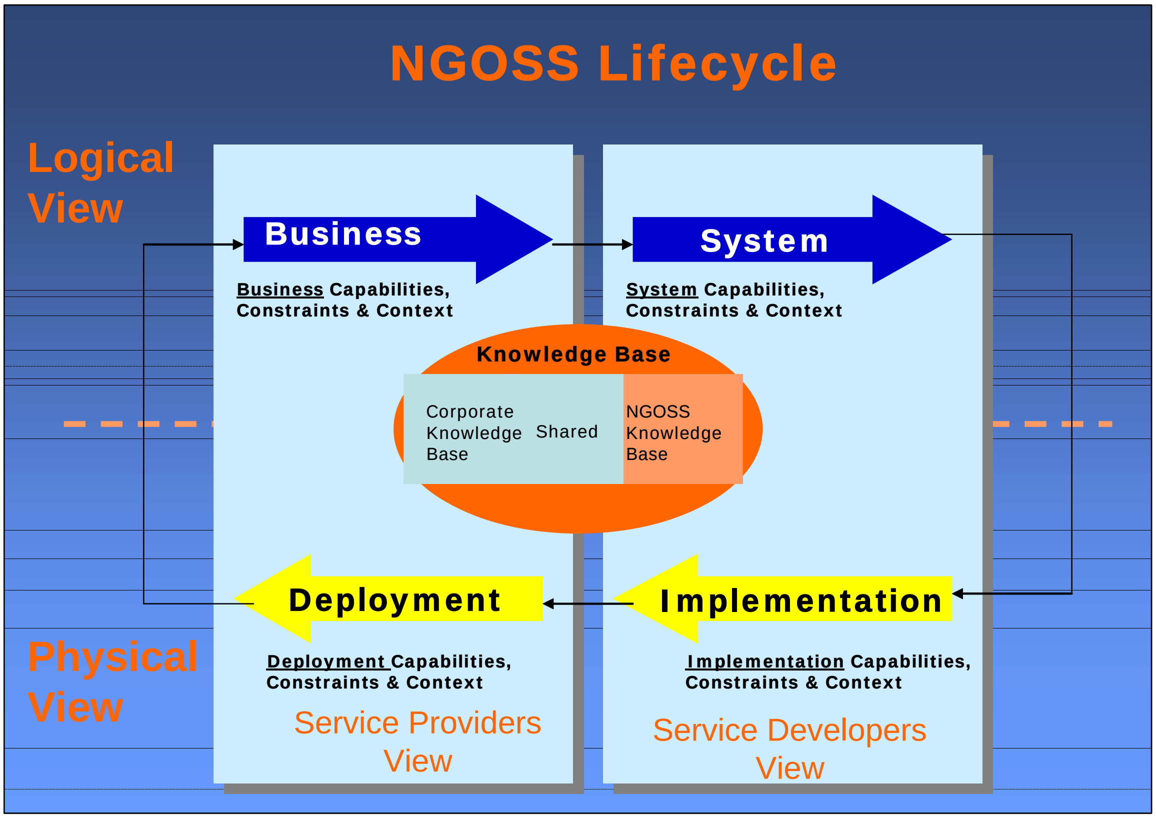


Figure 3.3. NGOSS Lifecycle: Business, System, Development and Deployment Views [TeleManagement Forum Lifecycle 2006]

NGOSS Lifecycle divided the two horizontal dimensions, the top dimension dealing with Logical Views of the system e.g. business problem, business processes, policies and the lower dimension dealing with Physical Views of implementation and deployment as depicted in Figure 3.3. The NGOSS lifecycle also depicts two vertical 'pillars' representing the interests of a Service Provider and Service Developer's view. The lifecycle identifies that the Business processes and policy as well as the deployment aspects are of primary concern to the Service Provider whereas the Service developer is more concerned with the system capabilities and constraints and the actual implementation. From this, the Lifecycle identifies the four ‘views’ of the system, namely the Business, System, Implementation and Deployment Views.

The NGOSS Lifecycle also identifies Knowledge Base. This comprises a Corporate Knowledge (of the Service provider) which represents accumulated experience collected from operating the business, NGOSS Knowledge which represents information, policies and process descriptions identified from NGOSS, and Shared Knowledge which is common to both NGOSS and a corporation.

The NGOSS proposes a five step Methodology called SANRR where SANRR represents ***S***cope, ***A***nalyse, ***N***ormalise, ***R***ationalise and ***R***ectify. The methodology suggests applying these five steps to each of the four Lifecycle view (Business, System, Implementation and Deployment). Table 3.1 describes the Purpose, Activities and Outputs associated with each step [TeleManagement Forum Lifecycle & Methodology].

STEP	PURPOSE	ACTIVITIES	OUTPUTS
Scope	• Defines solution boundary by understanding and documenting business purpose.	• Document the mission and goals of current business operations.	• Mission Statements • High Level Use Case(s)
Analyze	• Document existing operating environments & desired operating environment	• Identify Processes for improvement. • Identify Policies. • Identify Information/Data Models. • Define target Processes, Policies and Information Models.	• Detailed Use Cases. • Detailed Processes (with references to eTOM). • Detailed Policies Information Model(s).
Normalize	• To facilitate interoperation of different Physical Views of same Logical view. • To provide mappings and information model extensions where appropriate	• Map information models for Processes to a common reference model (TM Forum's SID)	• Single normalised information model
Rationalise	• To identify new processes, policies and functionality that map need to be developed. • Identify any duplicated (existing or anticipated) functionality.	• Perform duplication analysis • Perform Gap Analysis (between existing processes, policies and information models and target processes, policies and information models).	• List of duplicated functionality • List of functionality gaps in existing environment.
Rectify	• To supply new processes, policies and functionality to fill gaps identified in the Rationalise step. • To Modify pre-existing functionality (removing duplication and redundancy)	• Fill Gaps, • Build new functionality • Obtain new functionality, • Modify existing functionality, • Re-use/retire functionality so as to best meet Business needs	• New Functionality • Replicated functionality removed, • Updated Knowledge Base (Corporate & NGOSS).

Table 3.1 SANRR Steps

The TeleManagement Forum's lifecycle methodology is still quite immature and requires further development. Currently there are no published case studies or implementation research reports which have attempted to implement the methodology. Also detailed design workflows and activities have yet to be developed which are vital to support component and system development.

3.4.3 Survey of Telecommunication Development Methodologies

Several international research projects, funded under the EU Advanced Communications Technology and Services Program and EURESCOM, have investigated the area of development processes for telecommunication management. The EURESCOM project P.610 has performed case studies developing multimedia service management systems [Nesbitt 1998]. As with other projects, these case studies have made use of UML. These case studies provided examples of the application of UML use case diagrams capturing the requirements of management systems, and UML class, sequence and component diagrams to the design of these systems. The EU Advanced Communications Technology & Services (ACTS) research programme has funded many successful projects in this area. The ACTS project TRUMPET performed a case study using ODP viewpoints modeled using UML, for an inter-domain service management problem [Kande 1998]. They found that some UML could be used to represents ODP viewpoints, with use cases used for the enterprise viewpoint, class diagrams for the information viewpoint, component and sequence diagrams for the computational viewpoint and deployment diagrams for the engineering viewpoint. However later research has indicated that a lack of semantic accuracy in the meaning of some UML models have given rise to problems with UML ability to representing ODP computational objects [Romero 2005].

A more detailed study into development methodologies for service management was carried out in the ACTS project, Prospect [Prospect]. This project implemented a series of multi-domain management systems in phases over three years, with the aim to reuse and evolve components between phases [Wade 1998]. A development methodology, principally developed by this author, was followed which employed use case modelling, class diagrams, sequence diagrams, collaboration diagrams and component diagrams

[Wade 1997]. The process was applied to the analysis of multi-domain management processes and to the complete development cycle, from analysis to testing, of both single providers' systems as well as individual reusable components. The process also formed the basis of the ACTS Guideline on Design of Multi Domain Management Systems [Wade 1997a]. The methodology aimed to support the iterative application of the development cycle to these systems and components, as was required by the phased nature of the project. ODP viewpoints were initially used in this process, however problems were encountered with the separation between the information and computational viewpoints. Though this division of informational and computational viewpoints was seen as useful for documenting a completed system, the tools were not available to provide the strong traceable links between information and computational objects that are needed if the design is to be modified through multiple iterations. The separation between information and computational viewpoint was therefore diluted and systems or components were designed using class diagrams, component diagrams and sequence diagrams that mixed computational and information object types. This provided the designers with the flexibility they required to express the design models in the way that most closely represented the solutions to the various tasks required of the system or component. Figure 3.4 illustrates the design cycle and identifies the stages in the design and implementation process.

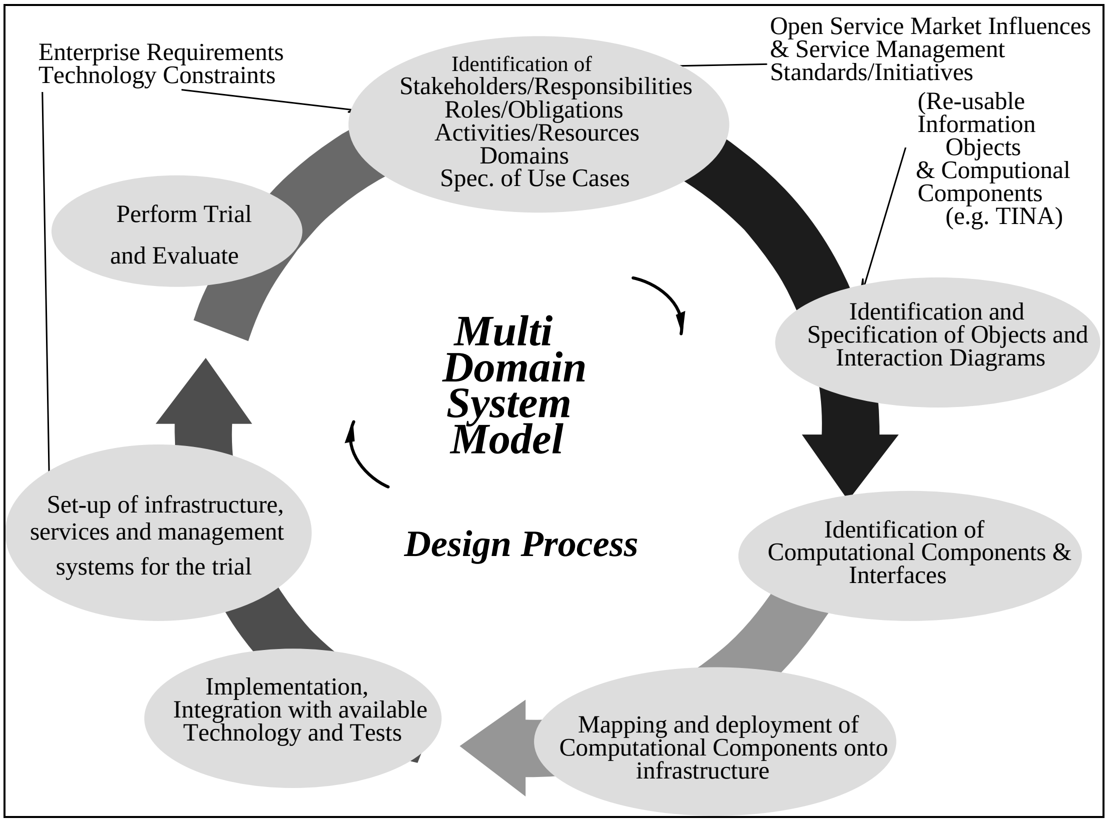


Figure 3.4: Prospect Design Process [Wade 1999]

The design process provided a structured way of developing, implementing and testing these object designs. It specified the design steps for

- developing multi domain business models (which includes the representation of organisations which are involved in the delivered managed service, assignment of responsibilities, identification of obligations and activities etc.);
- use case definition analysis;
- object identification and relation representation;
- definition of new computational components and the integration and extension of pre-existing computational components (e.g. from TINA C Service Architecture);
- distributed deployment of computational components;
- definition of platform architecture (nodes) and platform services;

- generation of test sets; and
- execution;

Prospect used UML to represent its use case diagrams, class diagrams, sequence diagrams and object collaboration diagrams. It also adopted conventions for the structure of use cases, for the naming of components and for OMG based specifications. The use of these notational conventions by the different groups, concerned with performing multi-domain analyses, system development or component development, made communication between the groups much more straightforward. A questionnaire of these developers revealed that the use cases in particular enabled the different groups to understand each other’s output more clearly.

3.4.4 Trends and Common Synergies

As telecommunication management becomes more embedded into modern, networked organisations, these overlap and influences between mainstream IT methodologies and approaches can only increase further the development of next generation telecommunications systems. However it is also clear that in an area dominated by industrial standards bodies and fora, any methodology for telecommunications management must be aware of the business, technical and regulatory influences which permeate the telecommunications management landscape. The use of RUP and its use of UML notation are now well established although heavily criticized as being very complex and causing confusion.

4 DEVELOPMENT METHODOLOGY: GUIDELINES FOR BUILDING BLOCK DEVELOPMENT AND BUSINESS PROCESS DRIVEN SYSTEMS DEVELOPMENT

4.1 Introduction

This chapter presents the motivation and rationale for the proposed MOdel Driven Development methodology, called MODD. It presents the scope of the methodology and introduces the key concepts which underlie the methodology. The chapter then presents the first MODD guideline, namely the Building Block Development Guideline, and illustrates its usage via the development of a case study based on the design of Assurance Management building blocks. The case study illustrates the various development workflows and design progression through the guideline. Finally an example Building Block Contract specification is presented which was designed using the guideline. Chapter five presents the second MODD guideline, which focuses on the development of telecommunications management systems using Building Blocks and business process representations.

The development and specification of the MODD methodology was entirely the work of the author. The author was responsible for the development and application of MODD methodology within the EU IST FORM Research Project. This provided a validation of the methodology as teams of developers from different telecommunications stakeholders applied the guidelines in the development of their own management services/components and systems. Thus the author did not develop the management services/components and systems in FORM, but rather assisted the development teams in the application of MODD. As part of this application of MODD, the author conducted evaluations of both of the guidelines to determine the success of the methodology.

4.2 Motivation for Methodology

The methodology tackles the twin challenges of designing reusable components and providing a component-based approach to the implementation of business process driven management systems. In tackling this challenge, the different stakeholders involved in the supply chain of telecommunication managed services are recognised – namely the producers of open interface standards, the providers of off-the-shelf management component software, the developers of management systems which use those components and the service providers who operate those systems. The requirements of these stakeholders vary but all must be accommodated by the methodology. For instance, Service Providers and Management System Integrators require that management solutions can be rapidly constructed and flexibly deployed at low cost through the reuse of software components. Component providers wish to supply this need in a manner that supports as wide a reuse market as possible.

The Methodology proposes a business process driven approach to the construction of management systems solutions from re-usable software component designs, which are termed Building Blocks (BB). However, management component providers need to be able to develop new BBs or enhance existing BBs. Such development activity requires guidance concerning the development and specification of BBs. Therefore MODD proposes a guideline to support the development of management BBs. Thus the MODD methodology is divided into two separate but consistent guidelines, namely:

1. The ***Building Block Development Guideline***: This guideline is intended for management component providers developing management BBs for reuse by management service providers or management system integrators.
2. The ***Business Process Driven System Development Guideline***: This is intended for use by system integrators who are developing management systems, based on business process analysis techniques, and who wish to use off the shelf management BBs.

4.2.1 Background: the MODD methodology

The MODD methodology was developed as part of a European Research project called FORM (Federated ORganisations Management) [FORM 2002]. The MODD methodology builds upon and enhances previous methodology research projects performed by this author which were conducted during the European research projects Prospect [Prospect] and FlowThru [FlowThru]. Elements of these earlier projects have been presented in Chapter 3.

The MODD methodology was used to develop telecommunication service management software components and systems as part of the FORM project. The author was responsible for researching, devising, specifying and evaluating the development methodology. The author used this application of the MODD guidelines, to validate and evaluate MODD. The author was responsible for the development of the guidelines in the FORM project and for ensuring the adherence to the MODD guidelines by the FORM partners (which were commercial and research software developers across eleven organizations). These organizations included network (device and software) vendors, telecommunication operators, system integrators as well as academic researchers. In order to illustrate the usage of MODD, examples of artifacts, developed by the FORM partners are presented. In particular, examples drawn from service assurance, and VPN management are illustrated in chapters 4 and 5.

The MODD methodology was used the FORM project as a case study both to define requirements for development of reusable component designs (which defined as BBs and Building Block Contracts in this thesis) as well as to define the requirements for developing telecommunication service management systems based on business processes (which reused a framework of pre-existing component designs). The FORM project adhered to the MODD methodology throughout several development iterations over a two-year period and developed a framework of reusable BBs covering specific management areas such as security management, virtual private network configuration, service accounting and service assurance. The project also used MODD to develop a number of management application systems for Web and Network service assurance system, a quality of service based billing system and a secure provisioning system for virtual private networks.

4.3 The MODD Methodology

The MODD methodology encapsulates the twin objectives of the Building Block Development Guideline and Service Management System construction using existing components. These objectives address the needs of those developing BBs as well as those developing management systems based on business process modelling but constructed using BBs. These objectives are elaborated in more detail in the following subsections.

4.3.1 Objectives and Scope of Building Block Development Guideline

The objective of Building Block Development Guideline is the development of re-usable management BBs Contracts and BBs. The Guideline not only provides advice as to how to model BBs Contracts, but also prescribes how such BB Contracts should be represented to ensure that the contracts could be reusable by other actors (i.e. actors not involved in the development of the Building Block Contract).

More specifically, the objectives of the guidelines are to:

- Guide the design activities in developing BB Contracts & BBs.
- Specify the development workflows required to design the Building Block Contract.
- Identify modelling notations and the models to be developed during each development workflow.
- Indicate the traceability of artefacts[1] developed across the development workflows.

[1] Reminder: An artefact is a piece of information that is created, changed and used by actors when performing development activities. An artefact can be a model, a model element or a document [Jacobson 2000].

- Prescribe sets of artefacts to characterize and communicate usage of Building Block Contracts.

The Guideline focuses on model driven development. Thus, the workflows defined in the guidelines focus on the modelling of UML artefacts and models necessary to capture the design of BBs and Building Block Contracts. It is not within the scope of the guideline to extend the current UML standard (UML v1.4). The guideline attempts to work within this UML specification in determining the development workflows and prescribing the appropriate UML model specifications, which characterise BB Contracts and BBs.

The guideline focuses exclusively on the developmental workflows rather than project management workflows or environment development workflows. Also, since the BB development guideline concentrates on the modelling aspects of BB and BB Contract development, it does not detail the programming/coding aspects of development or technology specific aspects (i.e. technology implementation decision to use EJB rather than CORBA etc.), and technology testing execution.

4.3.2 Objectives and Scope of Business Process Driven System Development Guideline

The objectives of this Business Process Development Guideline are:

- To provide support for a 'Business Process Driven' approach to management system construction from re-usable BBs Contracts
- To provide a development guideline which will allow management systems integrators to construct management solutions from Building Block Contracts.

The Guideline assumes the existence of catalogue(s) of Building Block Contract Specifications and the BBs which support them. It is expected that these catalogues can be generated using the Building Block Development Guideline. The catalogues are expected to have a functional overlap with the management solutions to be developed. Building Block Contract descriptions in the catalogues are conformant to the Contract Description template defined in the Building Block Development Guideline. Thus for each Building Block Contract, there are prescribed sets of models and a description template for describing a Contract.

The Guideline provides a development methodology from business modelling to system testing. The typical starting point for the guideline is a Management System Integrator wishing to implement business process(es) using MODD.

As with the BB development guideline, the Business Process Driven System Development Guideline focuses on Model Driven Development. Thus, the workflows defined in this guideline focus on the modelling of UML artefacts and models necessary to capture the design of management systems constructed from BBs and Building Block Contracts. The guideline does not attempt to extend the UML v1.4 standard. The guideline attempts to work within this UML specification in determining the development workflows and prescribing the appropriate UML model specifications, which characterize the intended system and its construction.

The guideline focuses exclusively on the developmental workflows rather than project management workflows or environment development workflows. It does not detail the programming/coding aspects of development or technology specific aspects (i.e. technology implementation decision to use EJB rather than CORBA etc.), and detailed technology testing execution. However, the guideline enables the generation of XMI descriptions of the control (business) logic, which captures the rules necessary for building block (and interface) integration. These XMI descriptions can be used as a basis for automated integration using a variety of technologies and approaches e.g. workflow engines, scripts.

4.3.3 Key principals underlying specification of Guidelines

The Guidelines have been developed with some clear underlying principals, namely:

- The Guidelines are to provide an agile, lightweight development process which focuses on the development of key management artefacts
- The Guidelines are developed for key stakeholders in the development process, namely component (or service) developers and component (or service) integrators.

- The Guidelines are suited to general purpose telecommunication service management. They is not intended for the development of real time management systems or low level network element instrumentation.
- Usage of the guideline does not assume experience in other methodologies e.g. RUP, but does require some experience of UML.
- The Business Process Development Guideline presupposes a set of defined Building Block Contracts from which selections can be made to support the implementation of the business processes under construction. i.e. the Business Process Development Guideline assumes the existence of a set of building block contracts in the same management (functional) area as required for the business process under construction.

4.4 Building Block Development Guideline

The approach taken in developing the Building Block Development Guideline was to use best practice in software development and add new workflows, model, artefacts and specifications to capture the necessary information for Building Block development. The Building Block (BB) Development Guideline is loosely based on the Rational Unified Process (RUP). Several of the RUP development workflows are generally applicable to software design e.g. business modelling, use case modelling etc. However, although generally useful, RUP does not support key modelling artefacts and design activities, which are fundamental to the guideline including the notion of Building Block Contract, management reference points. Appendix 1 provides a description of RUP, and a description of the relationship between the BB Development Guideline and the RUP development process. Thus the BB Guideline could be considered as an agile development process as it provides a lightweight, iterative approach to designing components.

The Building Block Development Guideline focuses on model driven development, and consists of two phases, namely Context Modelling Phase and Building Block

Development Phase (Figure 4.1). In the Context Modelling Phase, the objective is to model the overall vision within which the BBs are expected to reside. The workflows in this phase focus on the overall business modelling (some of which the BBs would ultimately support), requirements engineering, object analysis and design. The second phase focuses on the re-organisation of these models and designs as BBs and Building Block Contracts. This phase produces the models, artefacts and specifications needed to capture the design of the Building Block Contracts and the BBs, which support them. The development workflows defined in the Guideline are specified using UML activity diagrams, with accompanying textual explanations. To provide greater clarity a worked example is used to illustrate each workflow (and the design artefacts produced by these development activities).

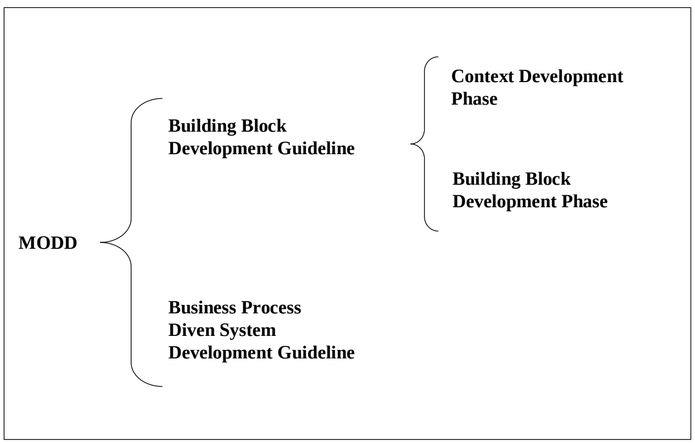


Figure 4.1 Overall MODD Structure

4.4.1 Building Block Development Guideline: Context Modelling Phase

This phase of the Guideline focuses on establishing the boundaries of the domain addressed by the Building Block development effort. This involves creating 'vision' document that outlines the scope of the management processes to be addressed by the BBs and their contracts during the development process. The vision document is key to the development work as it is the clear statement of the context and operational constraints of the domains in which the BBs are envisioned.

An initial reference architecture is then devised for the eventual BBs. This phase also involves the development of Business Model(s) and the initial Business Use Cases for the management process areas being investigated. The choice and scope of such management processes can be influenced by ‘standardised’ management processes e.g. TeleManagement Forum’s Telecom Operation Map. For example, if the envisaged BBs and contracts were to be related to the TeleManagement Forum’s standardization effort, then these processes may reflect some of the activities concerned with their prescribed Fulfilment, Assurance and Billing processes. The selection and customisation of such management process descriptions is dependent on the process areas of the BBs to be developed. Also during this phase, terminology and other relevant standards/models need to be identified which influence the emergent design e.g. IPDR for Accounting, DMTF CIM and IEFT QoS information models for Assurance.

Key outcomes of this stage are:

- Vision Document(s) indicating the scope, context and management business process areas of interest.
- Business Models identifying the Business Roles e.g. organisation(s) and Actors and where the management business processes reside.
- Initial Reference Architecture with reference points and domain boundaries.
- Requirements Capture & Management Document.
- Use Cases for the chosen management business processes. In addition to specifying the use cases, activity models representing the control and data flow involved in each use case can be modelled if required.
- Analysis Object Collaboration Model which can be used as a basis for identifying candidate BBs and Building Block Contracts in the next phase.
- Plan of how development work will proceed.

As the Guideline focuses on the development of BBs for Service Management, it will use the term ‘management processes’ to identify specific related functional areas rather than ‘management business process’. The term ‘business processes’ would also be appropriate but is less specific for the subject of this Guideline.

4.4.2 Process Workflows in the Context Modelling Phase

One of the fundamental challenges in developing BBs for management is to identify the appropriate aggregation of functionality, information resources and control logic. It is a huge task and a great challenge to attempt to develop such aggregations i.e. BBs and Building Block Contracts, from the outset. In fact, such ‘bottom up’ development of BBs is only possible where the functional domain has already been clearly represented and is well understood or where there are, pre-existing detailed models and designs available. This is not necessarily the situation as development effort required for the identification and construction of such re-usable BBs in telecommunication service management can be difficult. The Guideline uses a top-down approach to identify the candidate process areas, information and control objects required. Once the domain has been analysed, the actual analysis & design of the BBs can begin. Where extensive pre-existing design models exist, the context-modelling phase can be used to refine the existing models and identify where additional modelling is necessary.

This Guideline adopts a Business Model/Use Case Driven approach to represent the organisational domains and process areas of interest. The development activities and workflows to be carried out in the Context Modelling Phase are identified below.

1. **Perform Business Modelling Workflow –** This process workflow facilitates the definition of business model(s) based upon management business processes. This involves identifying Business Roles, Business information entities, Business Use Cases and Organisation Units.

2. **Define Reference Architecture Workflow** - Develop a Reference Architecture that identifies reference points[1] between organisational boundaries, the placement of

[1] A Reference Point is located at the boundary between two organisations. It is the point at which a business process traverses organisational boundaries. In telecommunication management, such ‘inter organisational

process areas within these boundaries and the relationships between these process areas across organisational boundaries.

3. **Perform Requirements Analysis Workflow –** This involves such development activities as Perform Requirements Analysis, Development of Use Cases and Supplementary Requirements Specification. As mentioned previously, this involves the modelling of activity graphs (diagrams) to represent the various control and data flows in the use cases.

4. **Develop Analysis Object Models Workflow –** This involves the development of analysis objects and development of analysis collaboration models. These analysis models are used in the Building Block Development Phase for identifying candidate BBs.

Figure 4.2 provides a simple graphic, which illustrates the top level of each of the workflows defined in the Context Phase. It also indicates the iterative nature of the activities in this phase.

---

domain' process interactions are important as they are the points at which public interfaces or protocols may be required between 'foreign' systems.

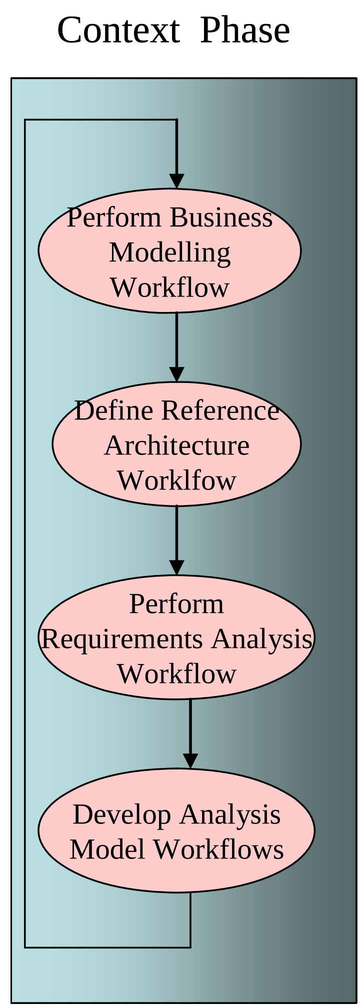
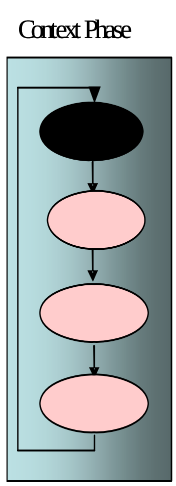


Figure 4.2 Context Phase Workflows

These activities and workflows are described in detail in the following section. Examples of models and artefacts developed using the methodology are also presented to illustrate the guideline's usage. These examples are drawn from the area of service assurance. An icon indicating which workflow in the context phase is being presented serves as a reminder of the overall flow of development activities in the Context Phase.



4.4.2.1 Perform Business Modelling Workflow

The objective of this workflow is to define key Business Roles, Business Use Cases, Business Processes and Business Resources/Information Entities for the functional domain of the intended BBs. The workflow defines a flow of activities which all the progressive development of the elements which ultimately form the Business Model and process model which the Building Block(s) are intended to address. There are two aspects from which the Business Modelling is performed. The first is the External View of the

Business, the second being the Internal View of the business within each organisation. The result of this workflow is the specification of the following models:

(i) Business Use Case Diagram(s): depicting business roles (workers and/or organisations) and use case name (external view)

(ii) Use Case Realisation, which models the business workers and entities/resources needed to carryout the use case (internal view)

(iii) Activity Diagram depicting the activities involved in carrying out the use case (internal view).

Figure 4.3 specifies the necessary development activities defined in this workflow.

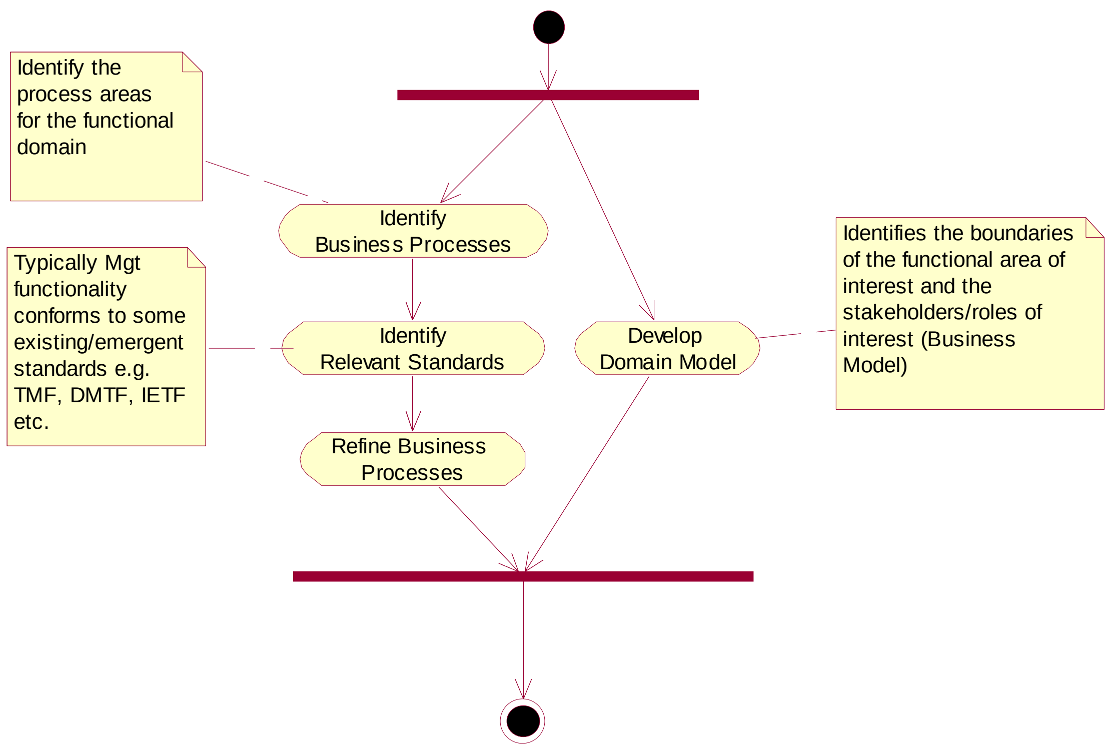


Figure 4.3 The Perform Business Modelling Workflow

Rationale and Commentary

The workflow starts by identifying the various business roles, which are to be represented in the envisaged system. These business roles are placed within the envisaged organization(s) to identify organisational boundaries. Interactions between the business roles are modeled as associations. Thus, the business roles and role interactions can be used as a basis for identifying business processes.

From these roles and associations, use cases (visable external from each organisation) can be identified and developed. Typically several key business use cases would be modeled for each interaction between business roles.

Business Use case realisation models can then be developed (i.e. views of what would happen internally in the organisation) for each (external) Business Use case. These internal Business Use Case Realisations model the business workers and entities/resources, which are needed within the organisation. Finally Business Activity diagrams are developed to refine the business activities descriptions, control flows and data flows within the organisations.

4.4.2.2 Example Building Block Business Modelling Workflow

In order to illustrate the development process better, a case study will be presented, which outlines the development activities for each of the workflows. The example is based on a generalized WWW based application service provider, a customer organization, multiple IP service provider and a ‘management service provider (in this case a service provider who monitors and assures the end to end delivery of the WWW based service offered by the application service provides. In the example suppose we are concerned with the development of ‘Assurance’ building block(s). Suppose we wish to develop one or more BBs and building block contracts to support the assurance of a WWW based information service (offered by the application service provider). More specifically, suppose this assurance set of BBs are required to assure the end-to-end operation of the WWW based service (from information source to end consumer). Also suppose this end-to-end service assurance potentially involves the consumer of the service being connected via one or possibly multiple Internet Service Providers. The scope of these building block(s) however, is that of monitoring and managing the Web based information service and not the underlying network connectivity. Finally, suppose this assurance service, is offered by an independent service provider (called an Assurance Provider).

The first activity, Perform Business Modelling, assists the designer in identifying the business context in which the building block(s) are expected to perform. The workflow prescribes the modelling of the Business Actors, Potential Business Organisations, and a set of use cases, which would be required.

Figure 4.4 depicts the Business Model, where the independent organization called the Inter Enterprise Service (IES) Provider, is providing the B2B service assurance between two enterprises: an Application Service Provider (ASP) and its Customer Organisation. The business model also shows the assumption of a Network Assurance role within the IES Provider. This role is responsible for dealing with intermediary Internet Service Providers (ISPs) to provide quality assured IP connectivity service.

In the model the principle roles within the various organizations are represented as 'business workers'. The relationships between these business workers are also represented to show, for example, the customer role being able to get a reporting service from the IESP.

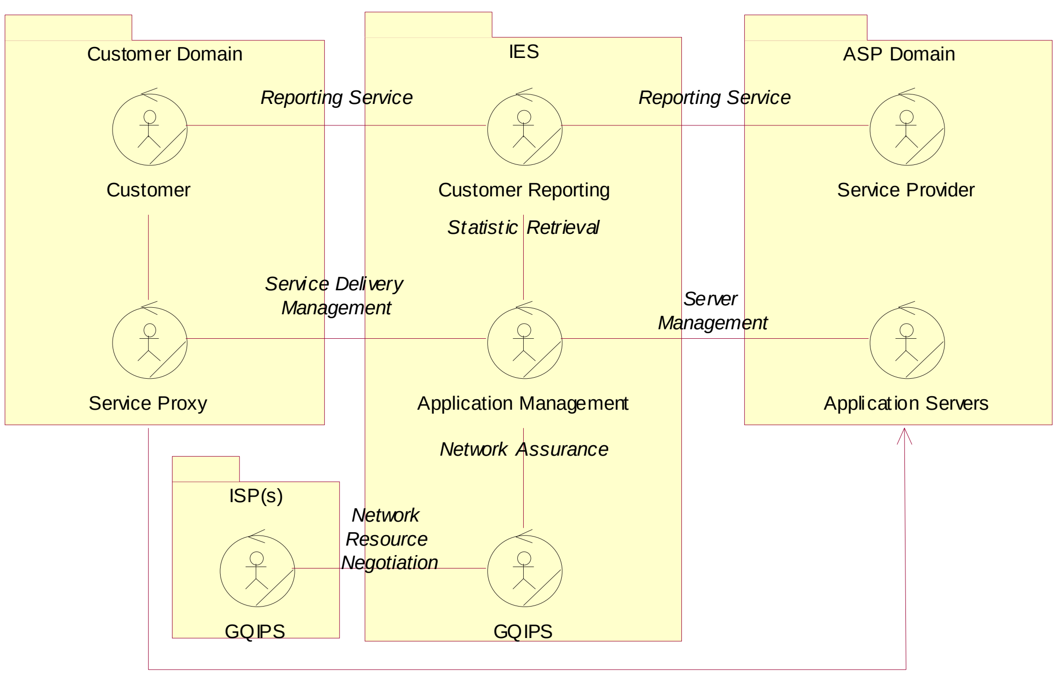


Figure 4.4 Business Model from Assurance

From this Business Model, actors and roles, an initial set of use cases can be defined. These define the services, which are being offered with regard to assurance of the WWW based information. The use cases for this assurance service are presented in Figure 4.5

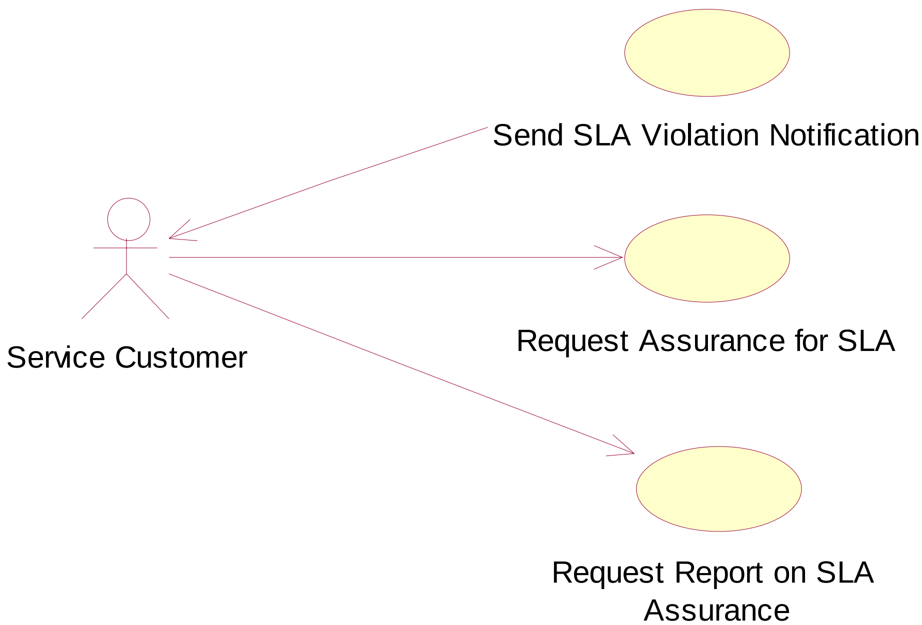


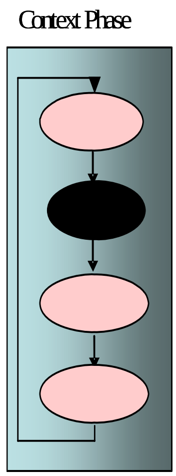
Figure 4.5 Business Use Case for Assurance

The use cases presented in figure 4.5 only show a subset of all the possible use cases in the Business Model. The use cases depicted focus on the allowed interaction between a customer of the assurance service, and the assurance service provider. In particular, the use cases depict (i) the customer requesting an assurance of the (application) service by specifying a service level agreement (SLA); (ii) the customer requesting a report on the performance of the (application) services in terms of the agreed SLA; and (iii) the service customer getting a notification that the SLA has been violated.

Thus the artefacts developed in this workflow are, Business Actors, Business Model and Business Use Cases. It is also possible to perform further business modelling by beginning to model the Business Activity Diagrams. These activity diagrams can be used as one of the inputs in the business functional requirements specification.

Also identified at this stage are the set of relevant standard bodies and their specifications e.g. DMTF & IETF standards for Assurance, TeleManagement Forum, IPDR for Accounting. In particular, the TeleManagement Forum has identified several business process areas, such as Assurance, Fulfilment and Billing. These process descriptions can be used either as a basis for the use cases and business activity diagrams, or just as a reference of typical telecommunication operator process requirements.

4.4.2.3 Define Reference Architecture Workflow



In order to provide a coherent logical structure through which the management business processes can integrate, a logical architecture is developed. It is useful to provide such a single diagram, which shows the logical separation of management processes, the organizational boundaries of the stakeholders and candidate reference points between those management processes. This helps communicating/explaining how the functional areas could co-operate, as well as providing a common 'map' around which the development teams can co-ordinate. The notion of 'reference point' is also important as these identify possible interactions across (inter-organisational) boundaries. Figure 4.6 describes the Define Reference Architecture Workflow.

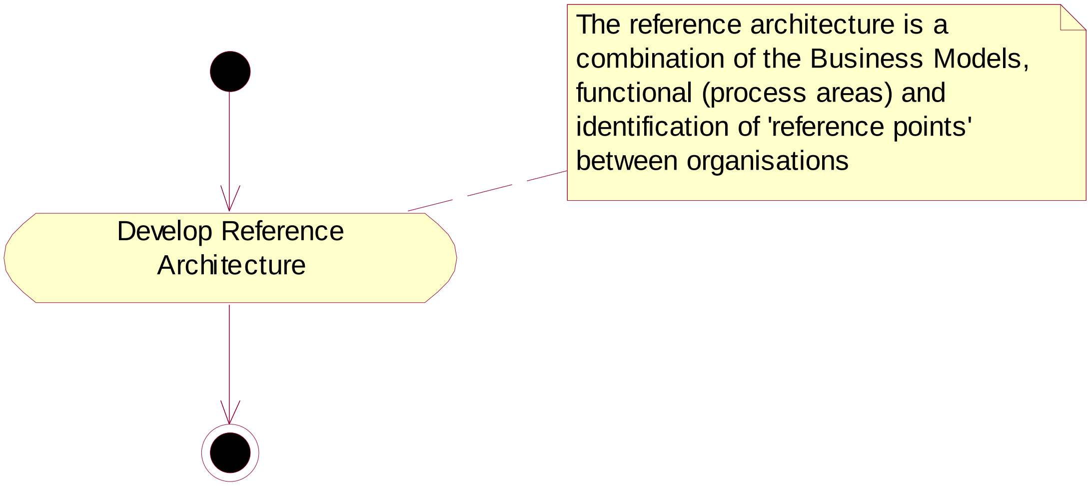


Figure 4.6 Define Reference Architecture Workflow

4.4.2.4 Example: Reference Architecture for Assurance

In the case study, the reference architecture identifies the process areas and reference points for our business model. Figure 4.7 presents a snapshot of the Reference Architecture for the stakehoders indicating a range of possible processes such as order handling, SLA negotiation as well as the assurance processes. The Reference Architecture can be revisited and refined several times during the guideline execution.

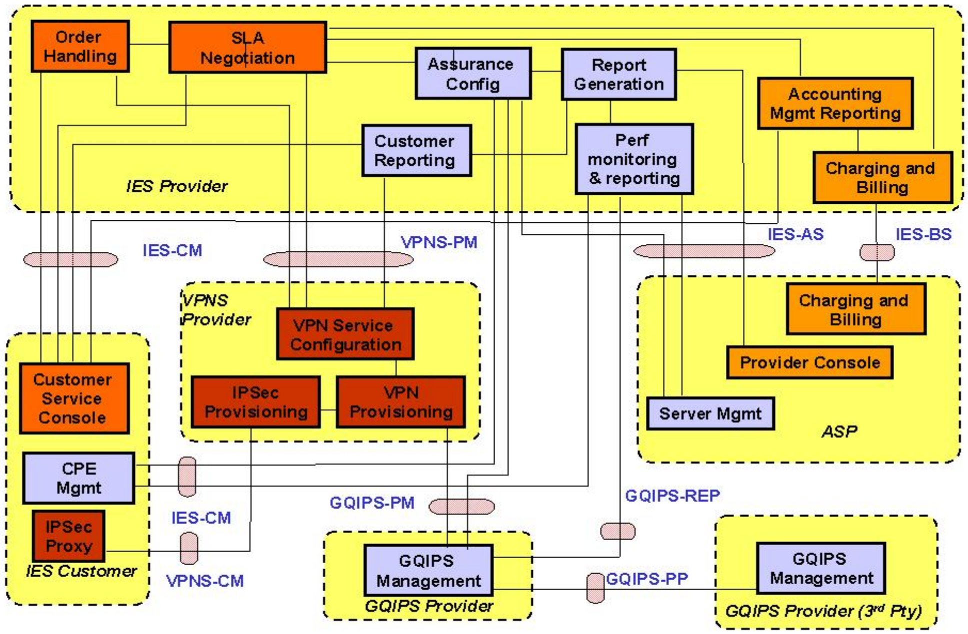
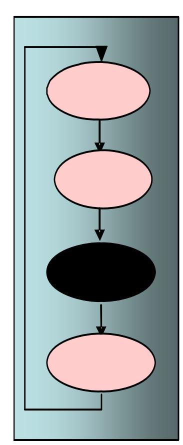


Figure 4.7 Initial Reference Architecture (in this case it's the FORM Reference Architecture)

In the development of the Reference Architecture (for potential Assurance Building Block(s)) it is necessary to identify the key interactions possible between the assurance processes in the different actors (organizations) as well as between other management processes e.g. Billing, Fulfilment. Also interactions which cross organizational boundaries can be grouped together to form Reference Points.

In Figure 4.7, the process areas of assurance are highlighted as assurance configuration (to set up the required assurance monitoring of a SLA), customer reporting, performance monitoring & reporting, and report generation. The architecture also identifies possible interactions between the process areas

Context Phase



4.4.2.5 Define Requirements Analysis Workflow

In order to identify candidate functionality and behaviour within the management processes, software requirement specifications and supplementary specifications are developed. Such requirements may be based on a market analysis of customers with regard to the functional areas. Other requirements may be gleaned from standards bodies and published requirement specifications e.g. TeleManagement Forum's requirements for management BBs [GB909 2001]. The use cases and the functionality identified within them, is at the 'system modelling' level (rather than the business modelling level). Thus the requirements modelling workflow is trying to identify functionality which needs to be supported by software under design, rather than identifying very abstract, high-level business activities. Figure 4.8 illustrates the development activities involved in this workflow.

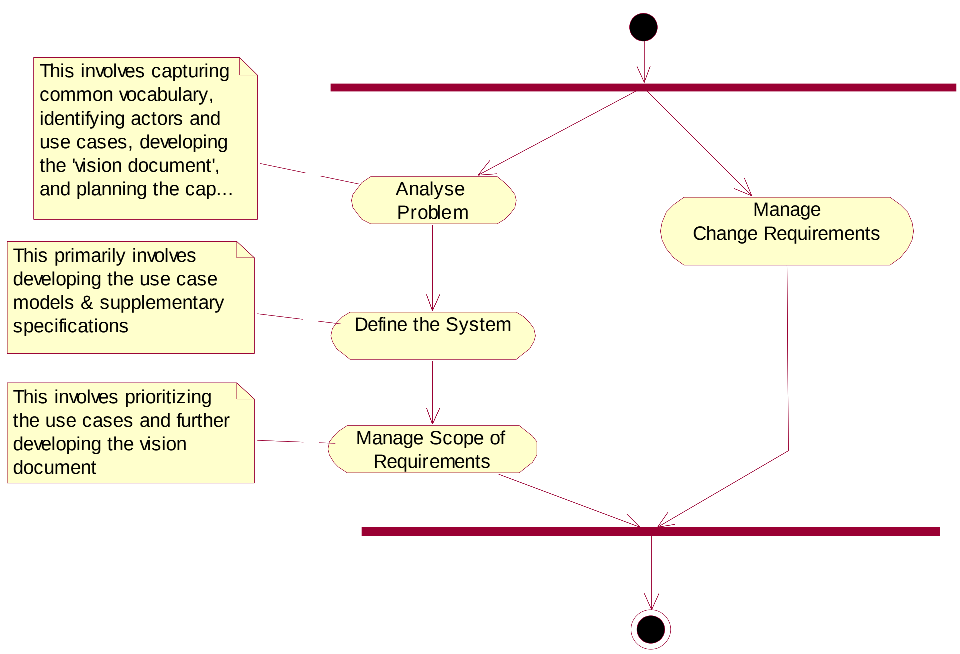


Figure 4.8: Define Requirements Analysis Model Workflow

During these development activities, Use Case Models are developed which describe the desired behaviour of the envisaged systems. Use cases at the boundaries in each of the functional areas are developed. These use cases provide both the actors (roles), which would make use of the management services, and a specification of each of these management services as a use case. The use cases consist of Use Case Model diagrams, supplementary specifications and activity diagrams representing the control flow between the activities. These development activities are customised from the RUP Requirements workflow.

As explained earlier, the end goal of the building block guidelines is to develop BBs within each functional area, but it is important to develop quite wide-ranging requirement sets and use cases to ensure breadth of coverage for each functional domain.

4.4.2.6 Example: Requirements Analysis Modelling

Suppose, in the case study, the intended BBs and Building Block Contracts we wish to develop are solely related to Assurance. This workflow helps to define the boundaries and actors, which would be appropriate for an assurance system. Figure 4.9 depicts the use cases actors, and boundaries of an assurance system and Figure 4.10 outlines some of the use cases modeled for the assurance functional domain. Figure 4.11 presents one of the activity diagrams associated with the Assurance Use Cases, namely the activity diagram for the Agree Assurance Support for SLA use case.

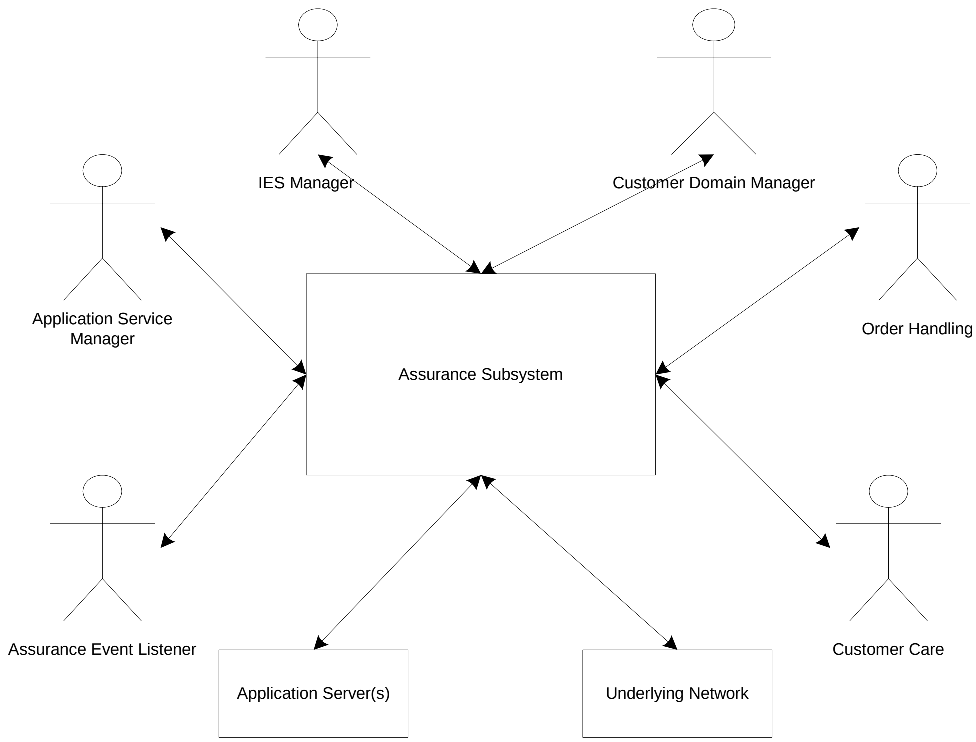


Figure 4.9 Use Case Actors & Boundaries

The above use case diagram identifies the different actors of interest for the assurance subsystem. These can be human actors e.g. application service manager, IES Manager or other systems e.g. assurance event listner. Each of these actors and their relationship(s) with the assurance subsystem are then expanded into use cases (Figure 4.10). The use cases identify individual functional interactions between the actor(s) and assurance system. Each of the usecases identifies a desired function of the assurance subsystem e.g. requesting a SLA (Service Level Agreement) report, Terminating assurance support for an SLA etc..

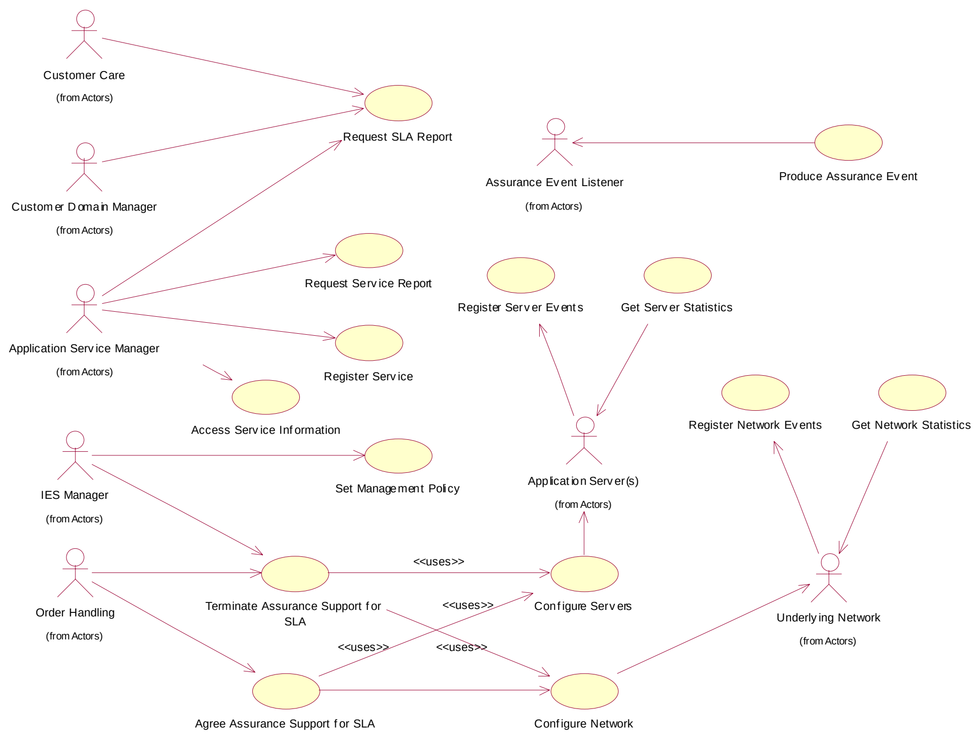


Figure 4.10 Use Case Models (Assurance)

Each use case is then expanded to describe the steps or activities which are needed to carry out the desired function. Figure 4.11 presents an activity diagram which captures the needed steps in performing the 'Agree Assurance Support for SLA' function. The steps indicated in the activity diagram are usually defined initially as a high level as the analysis stage (later) provides more detailed development.

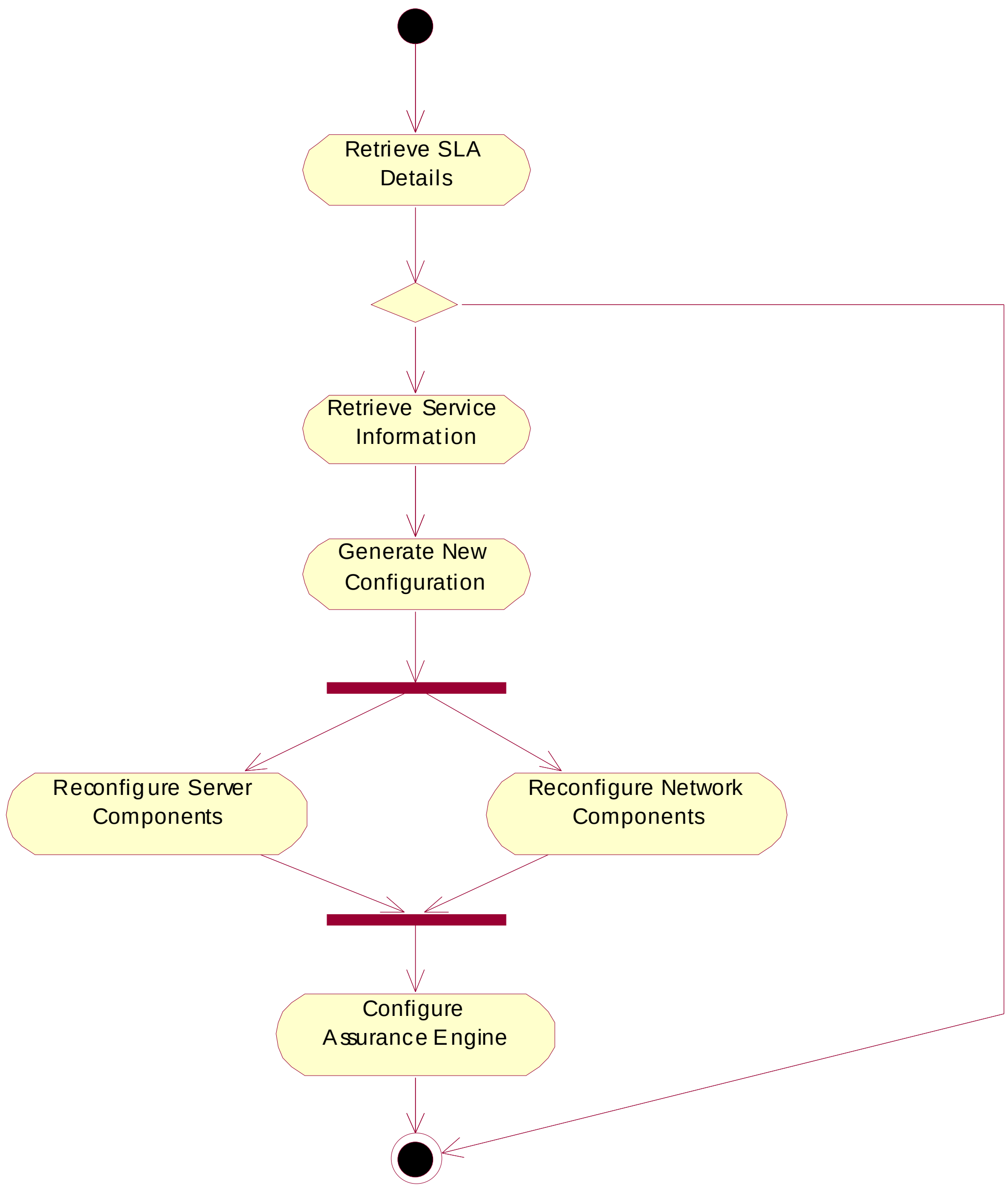
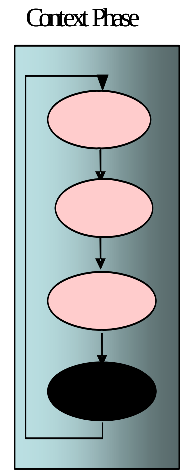


Figure 4.11. Activity Diagram: Agree Assurance Support for SLA

4.4.2.7 Develop Analysis Models Workflow



This development workflow focuses on the identification of analysis classes and their interactions based on the use cases defined earlier. Artefacts developed during this workflow include Design of Analysis Classes, Collaboration & Sequence Diagrams and Interfaces. These artefacts can be brought together into subsystems. These subsystems can be thought of as a logical collection of classes, which may be useful in forming potential BBs. Initially these analysis classes can be identified from the Use Cases and activity diagrams developed earlier. In this workflow an initial Information Object Model (captured as a class diagram) is formed. The development workflow is defined in Figure 4.12 below.

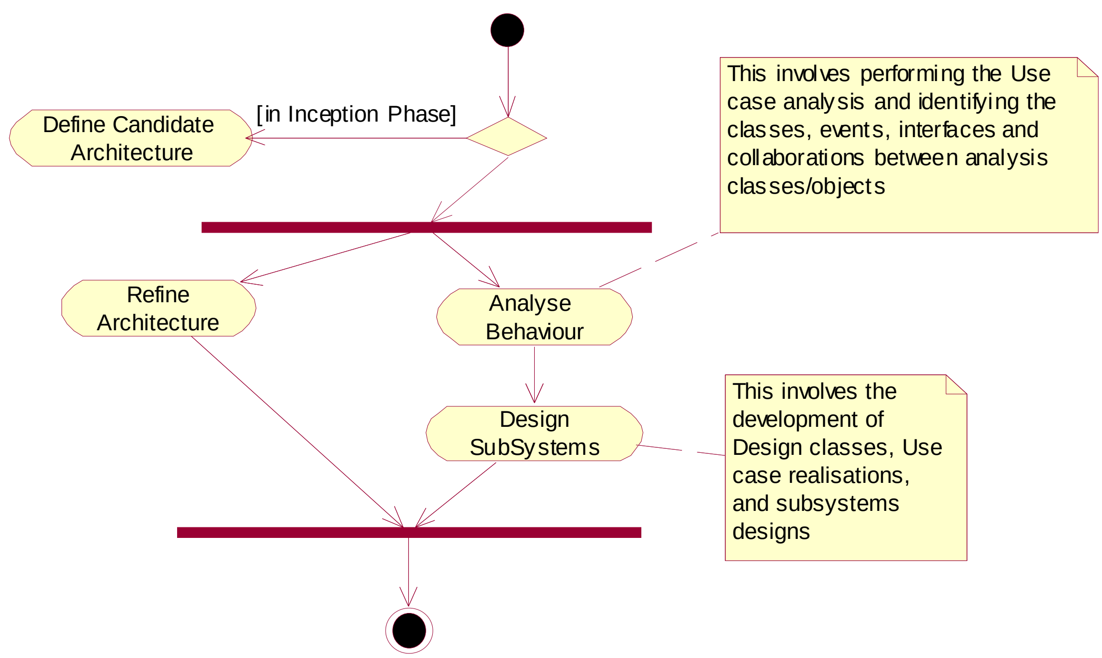


Figure 4.12: Develop Analysis Model(s)

4.4.2.8 Example: Analysis Object Modelling

In our example case study we examine the analysis objects involved in the termination of an Service Level Agreement (SLA) use case. The workflow identifies the relevant analysis objects needed to carryour the use case e.g. order handling, configuration assurance etc as depicted in Figure 4.13. The analysis objects can be sequenced to support the various use cases defined earlier. An example of such sequencing is described Figure 4.14.

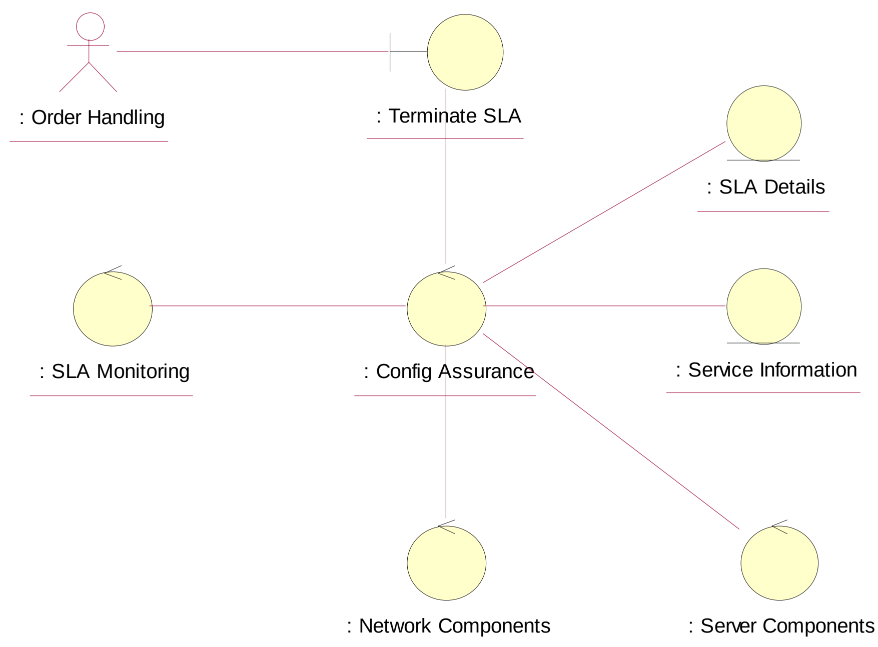


Figure 4.13 Collaboration Diagram: Agree Assurance Support for SLA (Analysis Model)

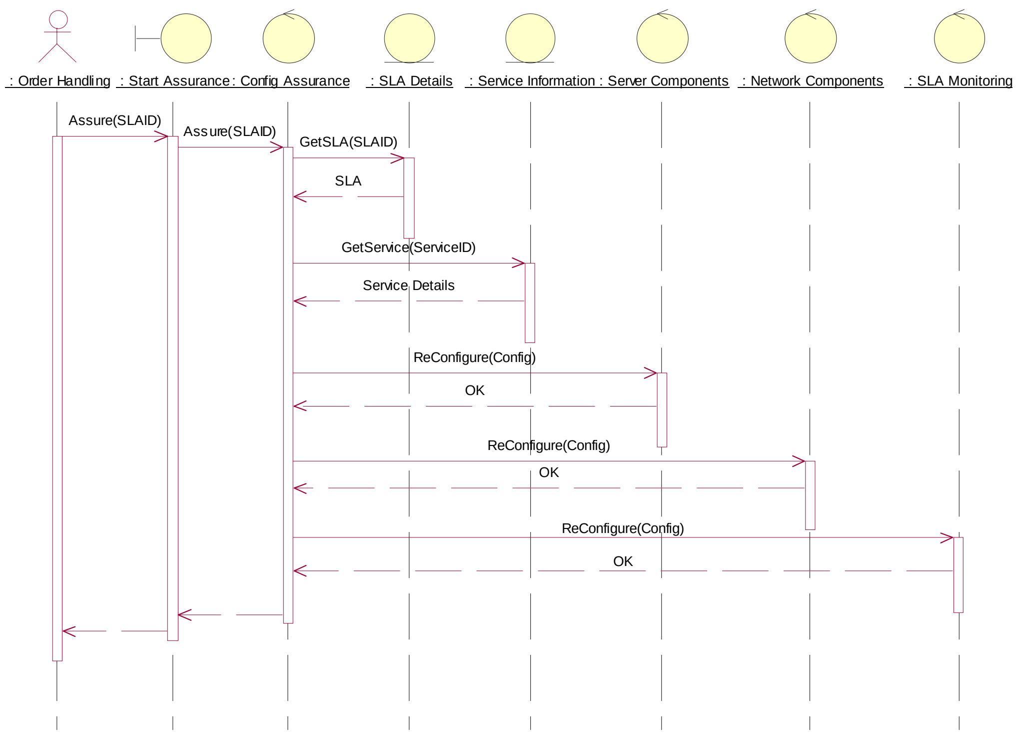


Figure 4.14 Interaction Diagram: Agree Assurance Support for SLA.

4.4.3 Iterating the Context Phase Workflows

Although in the example we have presented quite detailed models for each workflow, it is usual that these models are arrived at only after a number of iterations of the content phase. This is particularly true in defining and refining the reference architecture and business models. Thus this phase requires several iterations in order to develop a reasonable understanding of the context within which the intended BBs are to be designed. These iteration provide a deepening of the context by refining the business models and use cases which may surround the BBs. However, it is important to note that not all aspects of the Business models need to be fully modelled in this phase, as the phase is only intended to capture the context for the BBs, and not necessarily intended to provide a entire system development.

4.4.4 Building Block Development Guideline: Building Block Modelling Phase

This Phase focuses on the revision and refinement of the models and Vision Document(s) completed in the Context Modelling Phase. In this Phase, the Reference Architecture is solidified. Also during this phase, potential BBs and Building Block Contracts are identified and the Use Cases to be supported by the BBs are modeled and refined. The descriptions of Building Block Contracts are further refined and Building Block Contract templates is populated. A Building Block Contract template defines the essential artefacts, including UML models, required to characterize and describe the BBs Contracts.

The main artefacts of the Building Block Modelling Phase are:

- A stable system architecture model.
- Development models for BBs Contracts.
- Development of models for BBs

4.4.5 Process workflows in the Building Block Modelling Phase

The workflows in the Building Block Modelling Phase include those of the earlier phase, as the overall development process is iterative. However, the workflows Perform Business Modelling, Define Reference Architecture, Perform Requirements Analysis, and Develop Analysis Object Models are re-iterated to refine the relevant artefacts. The workflow is 'architecture centric' in that it focuses on the (incremental) development of the BB and BB Contract Models (Figure 4.15). To achieve this an extra workflow in this phase is the Re-organisation of Analysis Objects into BBs.

Where multiple BBs and BB Contracts are being designed, it is possible to instantiate this phase for each of the BB and BB Contracts, or (more commonly) to provide different instantiations of the phase for different (closely related) BBs and BB Contracts. For example, if there were BB and BB Contracts to be developed in the areas of, for example, quality of service as well as provisioning, it would be possible to iterate through the development phase separately for each of the two areas. In this way, each instance of the phase would focus on a single, of a set of highly related BBs and BB Contracts. If such a separation occurs, it is important to maintain consistency of shared information (across the separations) and also to ensure consistency with the overall business mode and reference model. However, as the objective in this phase is NOT to develop a total system, but rather to develop sets of BBs and BB Contracts, this separation of development activity is acceptable.

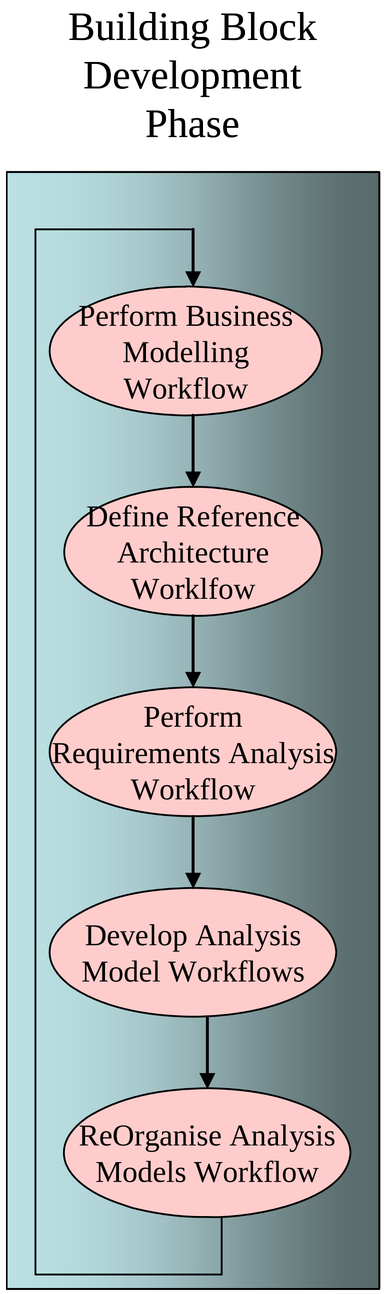
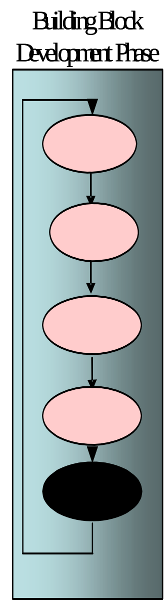


Figure 4.15 Building Block Development Phase Workflows

Model (Re)Organisation Workflow: This workflow identifies the development activities involved in Re-organising the Analysis Classes/Models in order to group useful behaviours/entities into potential BBs. One reason for this re-organisation may include the decision to adopt/use standard information models or to suggest interfaces and information models (e.g. IPDR for Accounting, CIM for Assurance). Another reason for re-organisation is that the analysis work up to this point, has mainly been performed to identify candidate system-wide functionality & information entities, and the development activities now focus more explicitly on BBs and Building Block Contracts. The design workflows suggest several possible criteria, which are useful in identifying potential BBs (or candidate component designs).

4.4.5.1 Re-organise Analysis Model(s) Workflow



This workflow is concerned with the identification of candidate BBs from amongst the analysis objects identified in the earlier development activities (Figure 4.16). Essentially a Building Block is an specification of a component design. BBs can support multiple interface types and multiple instances of those types. A Building Block Contract specifies a grouping of information and behaviours, which can be re-used to support management business processes. One or more Building Block Contracts can be supported (implemented) by a Building Block.

The notion of a Building Block and Building Block Contract differs from the notion of 'component' in some development processes (e.g. RUP), in that the Building Block can support one or more contracts and is described as a package of modelling artefacts. For example, in RUP, the term component only has meaning in the deployment model and not in the analysis/development activities. Thus, the closest RUP notion to Building Block is that of 'subsystem' which can be a collection of related development artefacts representing some functionality. However, the definition of a Building Block is much more specific and prescriptive than that of a 'subsystem' in RUP.

A Building Block description includes use cases and collaboration diagrams to indicate usage scope of the Building Block, a contract interface specification and a specification of information classes passed into or out of the BBs. A Building Block supports one or more contact specifications. Figure 4.16 describes the design workflow to support the design of a BB and a BB Contract.

The guideline suggests some criteria which may be useful in identifying a BB.

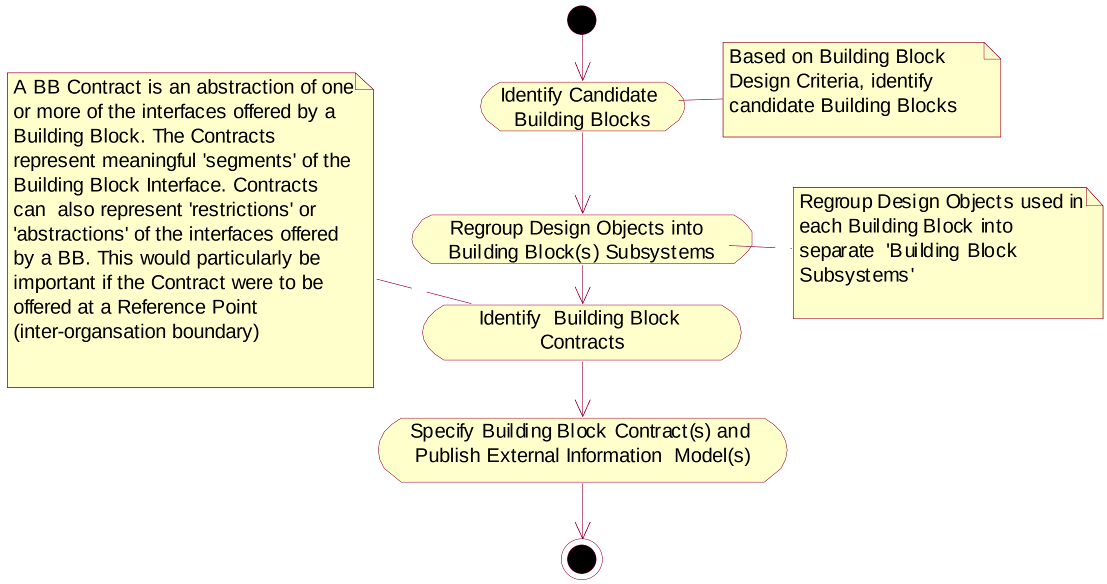


Figure 4.16 Reorganise Analysis Model Workflow

Criteria for scoping or determining Building Block

A non-exclusive set of criteria is defined to assist in identifying candidate BBs by reorganising the analysis classes identified in the previous development activity:

- Does the grouping of classes provide an Enterprise Wide information service, re-usable business logic or generally useful User Interface (i.e. at the Enterprise Information Tier, Process Automation Tier and Human Interaction Tier)?
- Does the grouping of classes represent some self-contained behaviour (logical grouping of closely related behaviours)?
- Is the level of inter-dependence between a set of classes (collaborating classes based on original use cases) sufficient as to suggest their close dependence?

- Is there a definite 'service' or 'services' that a group of classes can uniquely support (does it add a useful, distinct, service to the system)?

The guideline suggest the following (non exhaustive) list of criteria.

(i) As a potential unit of deployment in determining the granularity or boundary of a BB

(ii) As a service that BB could uniquely support (in determining the granularity or boundary of a BB)

(iii) By the degree of self contained behaviour in determining the granularity or boundary of a BB

(iv) By the potential of BB as a unit of manageability in determining the granularity or boundary of a BB

(v) As supporting one (or more) of the three tiers (presentation, application/business logic, persistence) when determining the granularity or boundary of a BB

(vi) By the level on inter-dependence between grouping of classes in determining the granularity or boundary of a BB

4.4.5.2 Example: Reorganising Analysis Models for modelling Building Blocks

In the case study the analysis objects are regrouped and remodeled into packages (represented diagrammatically as folders). The result of this reorganisation of classes into logically independent packages identifies candidate BBs.

This reorganisation can be based on shared information requirements, shared objectives, or the need for close collaboration. However, it is important to note that the classes within a single package need not be all of the same type i.e. some may provide persistency, others represent business logic or control objects. If we continue our case study concerning the assurance BBs, we can see that the analysis objects can be gathered into four related packages that provide service monitoring, assurance configuration, performance monitoring and report generation. Figure 4.17 identifies several packages (candidate BBs) each containing objects.

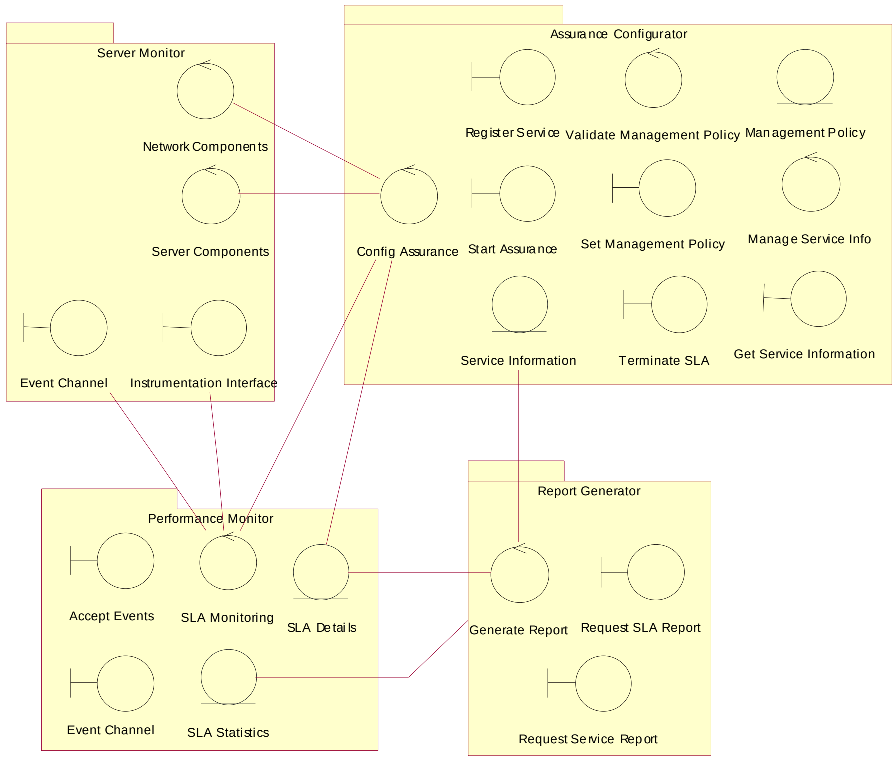


Figure 4.17 Grouping of Analysis Object into BBs

Sequence diagrams can also be modeled to show how the candidate BBs can be sequenced to support use cases identified earlier in the guideline (Figure 4.18). Note that the sequence diagram depicts interaction between the BBs and not the internal analysis objects.

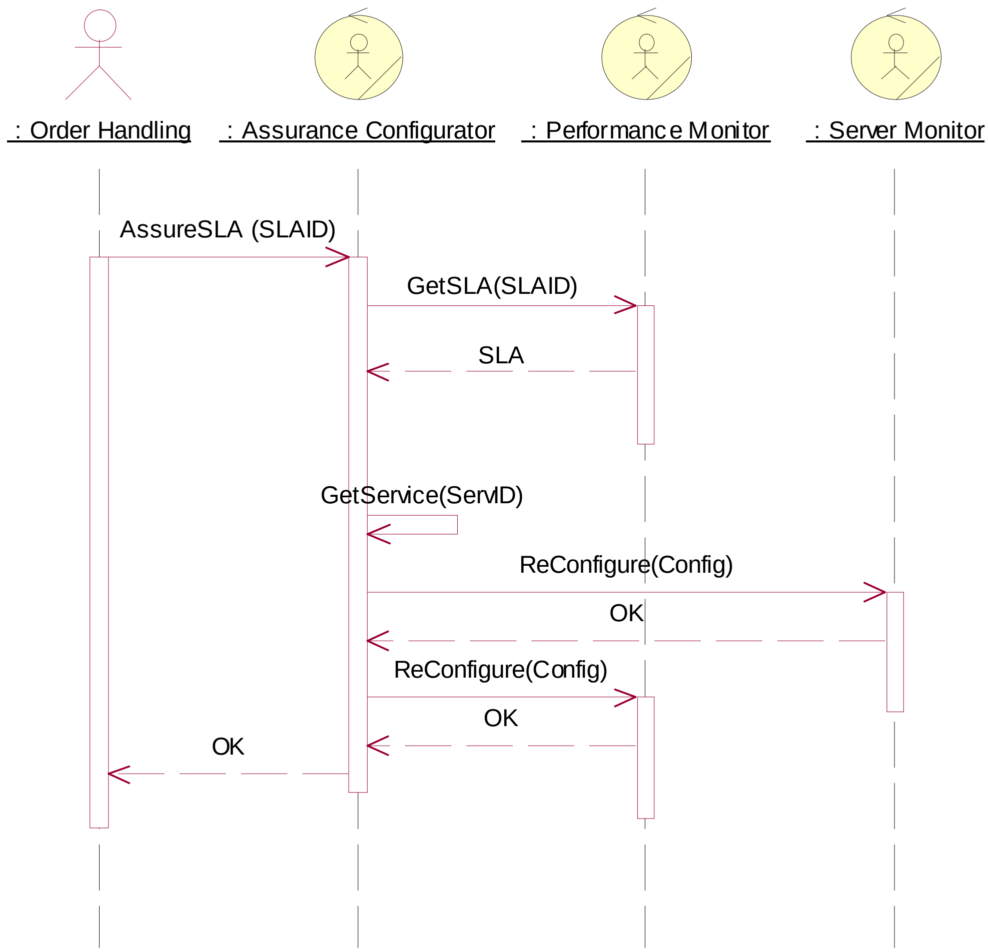


Figure 4.18. Interaction Diagram: Agree Assurance Support for SLA

4.4.5.3 Modelling Candidate Building Block Contracts

The template for describing a Building Block Contract is as follows:

- A Building Block Contract Name (specified as text).
- The names of defined Reference Points Supported by the Building Block Contract (if any) and the Business Role supporting the services provided by the Contract. These are points on the abstract organisational boundaries through which the Building Block Contract may be accessed.

- Contract description defining service offered by Building Block Contact (specified as text).
- Use cases & collaboration diagram(s) to illustrate usage scope of Building Block Contract
- Contract interface specification.
- Information Objects communicated at the interface of the Building Block Contract i.e. UML Class Diagram of information objects exchanged by the contract. This is called the Boundary Information Model for the Building Block Contract
- Technological description for Building Block Contract (specified as text).
- Collaboration diagrams illustrating the Building Block Contract potential interactions with other Building Block Contracts in the FORM Framework. The inclusion of these collaboration diagrams is optional as they are intended only to indicate where close reliance or relationships exist between Building Block Contracts.

There are two possible approaches to the specification of contracts. The first is to design the Building Block and then design the contract specifications, which that building block can support as a set of abstractions on the building block interface (as indicated in the activity diagram earlier). It is important to note that the contracts can offer different functions/interfaces or can support restrictions (or abstractions) on the interfaces supported by the BBs. An alternative approach to defining Building Block Contracts is to attempt to design the contracts first and then define the BBs to support such contracts.

A full example Building Block Contract specification and description is presented later in section 4.5.

4.4.6 Building Block Development Guideline: Building Block Implementation

The Building Block Development Guideline is focused on prescribing the workflows and modelling artefacts needed for BB Contract and BB development. The actual implementation and coding of the BBs are outside the scope of the guideline. However, many commercial UML based development tools provide code generation facilities. These facilities can generate skeletal code in a variety of programming languages and middleware technologies. The use of such tools and the resultant programming are not part of the guideline.

4.5 Case Study: Example Building Block Specification

In order to achieve a better understanding of BBs and their Contracts, this section presents the specification of a building block, which was developed for the FORM Assurance Domain. The case study example illustrates a Building Block Contract specification and the UML models used to describe them. It is important to note that although, in this specification, only a single contract interface is specified, multiple interfaces are also permissible for a Building Block Contract.

4.5.1 Specification of QoS Server Monitor Building Block Contract

BB Contract Name: Server Monitor

Reference Points: IES-SM

Contract Description:

This contract allows access to the CIM information base stored in the Server Monitor building block. The building block monitors server statistics, calculating secondary combinatory statistics when necessary. Both primary and secondary statistics are stored within the information base for retrieval. Objects facilitating the management of the Server Monitor itself are also present in the information base. These objects perform a number of different tasks such as initialising and managing downloadable extensions to the module.

4.5.1.1 Example: QoS Server Monitor Use Cases

The use cases should define the services offered at the systems boundary as well as the actors who would use the management service. Figure 4.19 identifies two assurance management services, namely start server monitor, and get server monitor statistics. These management services are depicted being used by the workflow engine (in the service assurance trial system, the various assurance business processes are initiated and integrated by a workflow engine) and a performance manager (another building block in the system).

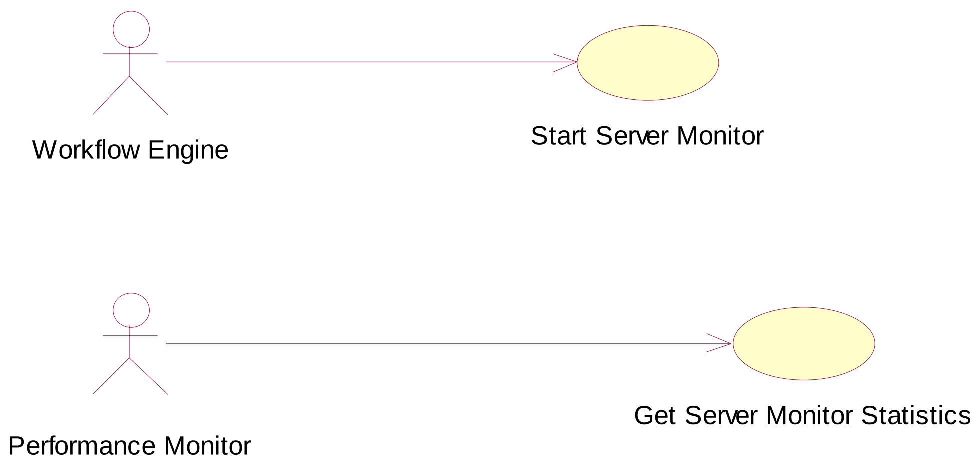


Figure 4.19 QoS Server Monitor Use Cases

4.5.2 Example: Building Block Contract Interface

The contract consists of the interface shown in figure 4.20 below.

The cimOperation parameter should contain a complete XML document that conforms to the “CIM Operations Over HTTP” standard specified by the DMTF. This standard specifies the structure of an XML document used to query and otherwise manipulate the CIM information base. It also specifies the format of the XML response to these requests (i.e. the returned string). The final item to note is that this method can also throw an exception to indicate that the cimOperation did not conform to the CIM DTD and therefore could not be processed.

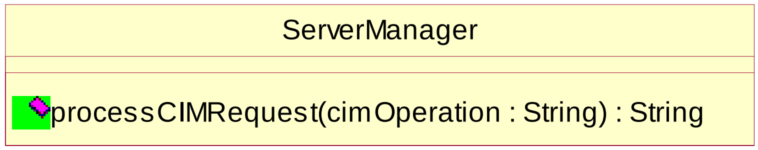


Figure 4.20 - UML Diagram of BB Contract Interface

4.5.3 (Building Block) Boundary Information Model

The external visible information model (termed the Boundary Information Model) supported by the building block contract is in the form of CIM classes and objects. It is therefore important to understand the structure of these classes and how they relate to each other. The Boundary Information Model for this Building Block Contract can be logically divided into four aspects the Server Monitor Management Information Model, Server Monitor Configuration Information Model, Service Description Information Model and the Calculated Statistics Information Model (as depicted in Figure 4.21).

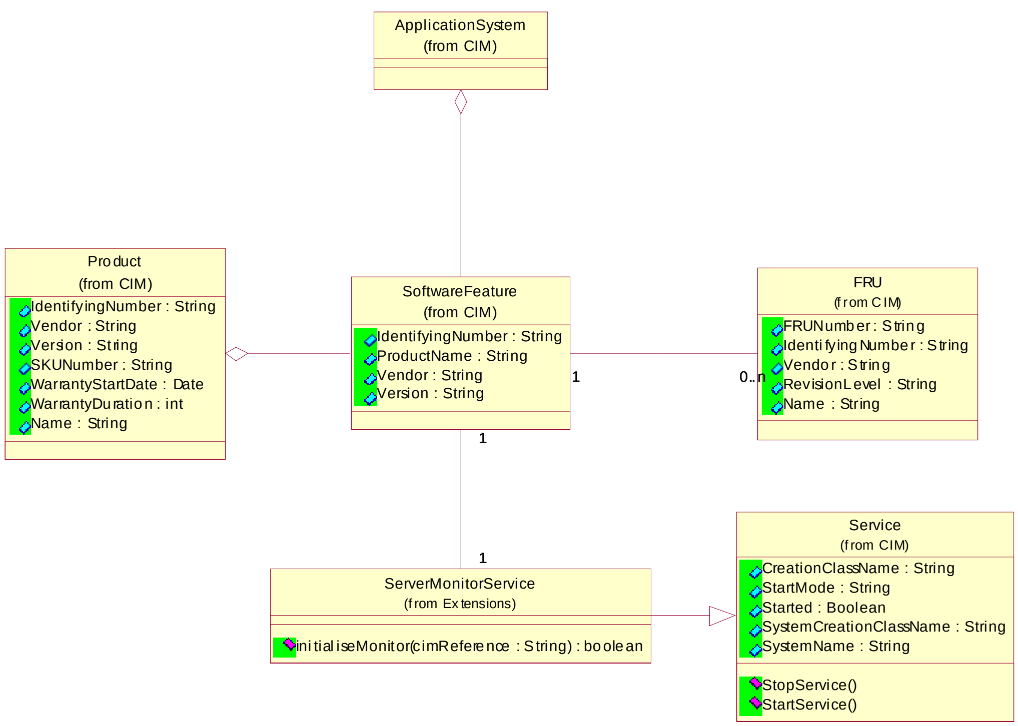


Figure 4.21 Server Monitor Information Model

Associated with each Building Block Boundary Information Model is an explanation of the classes in the building block.

Class	Usage
CIM_ApplicationSystem	Maintains high level information about the Server Monitor such as contact information etc.
CIM_SoftwareFeature	Describes the features of Server Monitor. There must be one instance called “Statistical Calculation”. Other instances may be made as deemed necessary.
CIM_Product	An instance of this class may or may not be provided. If provided it’s only purpose is to represent how certain software features comprise a product.
CIM_FRU	Associated with the “Statistical Calculation” instance will be zero or more instances of CIM_FRU. Each of these instances will represent a piece of code downloaded to the component to aid in the calculation of statistics.
FORM_ServerMonitorService	One instance of this class must be associated with the “Statistical Calculation” instance. This class provides the method by which the Server Monitor is initialised and eventually stopped.

4.5.4 Collaboration Diagram of Server Monitor BB with other FORM Framework BBs

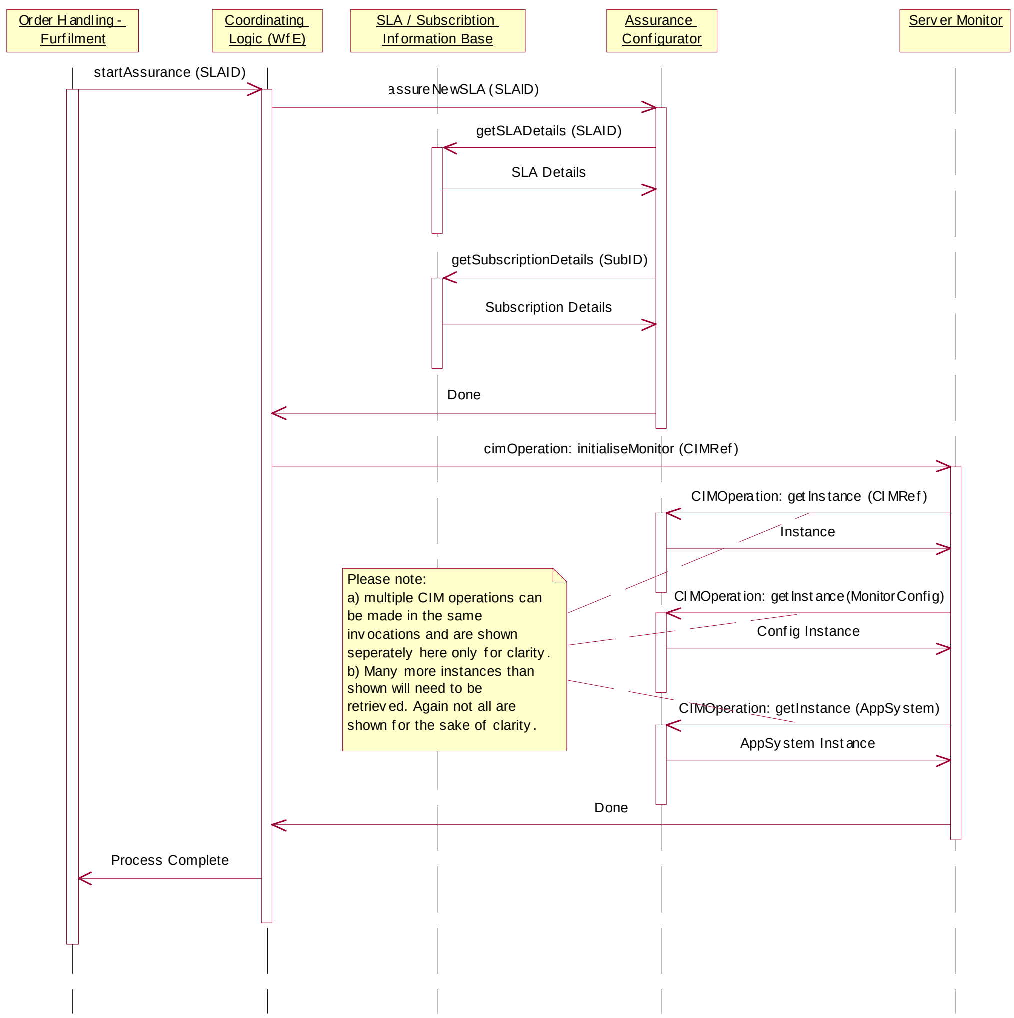


Figure 4.22 Server Monitor Collaboration diagram

The following is the XML specification of a Contract specification from the Server Monitor BB as specified by one of the FORM development team (B Cullen):

- <BBContract name="cs.tcd.ie/FORM/ServerMonitor" contractSpecifier="Brian Cullen" date="8/8/2001" version="1.0"

```
xmlns="http://www.ist-form.org/BBContractDescription"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ist-form.org/BBContractDescription C:\College\FORM\BB-Contract-Description-v2-1.xsd">
<description>This contract allows access to the CIM information base stored in the Server Monitor building block. This building block monitors server statistics, calculating secondary combinatory statistics when necessary. Both primary and secondary statistics are stored within the information base for retrieval. Objects facilitating the management of the Server Monitor itself are also present in the information base. These objects perform a number of different tasks such as initialising and managing downloadable extensions to the module.</description>
- <supportedReferencePoints referenceArchitectureURI="http://www.ist-form.org/ReferenceArchitecture">
  <referencePoint name="IES-CM" />
  <referencePoint name="GQIPS-PM" />
  <referencePoint name="IES-AS" />
</supportedReferencePoints>
- <contractScope>
  - <useCaseSet diagramURI="UseCase01.html">
      <useCaseCollaboration diagramURI="UseCaseCollaboration01.html" relatedUseCase="Configure Monitor" />
      <useCaseCollaboration diagramURI="UseCaseCollaboration02.html" relatedUseCase="Get Statistics" />
```

<useCaseCollaboration diagramURI="**UseCaseCollaboration03.html**" relatedUseCase="**View Configuration**" />
</useCaseSet>
</contractScope>
- <interfaceInteractions>
<interfaceInformation description="**CIM Operations Format**" fileURI="**InterfaceInformation01.html**" />
<interfaceInformation description="**EJB CIM Interface**" fileURI="**InterfaceInformation02.html**" />
<interfaceInformation description="**CIM Service Description**" fileURI="**InterfaceInformation03.html**" />
<interfaceInformation description="**CIM Server Monitor Settings**" fileURI="**InterfaceInformation04.html**" />
<interfaceInformation description="**CIM Server Monitor**" fileURI="**InterfaceInformation05.html**" />
<interfaceInformation description="**CIM Calculated Statistics**" fileURI="**InterfaceInformation06.html**" />
</interfaceInteractions>
- <boundaryInformationModel>
<informationModel modelURI="**BoundaryInfoModel01.html**" />
<informationModel modelURI="**BoundaryInfoModel02.html**" />
<informationModel modelURI="**BoundaryInfoModel03.html**" />
<informationModel modelURI="**BoundaryInfoModel04.html**" />
</boundaryInformationModel>
<technologyDescription>**All information accessed and passed through this contract is done so in CIM format.In particular this contract supports the "CIM Operations over HTTP" standard specified by the DMTF. This standard specifies the**

structure of an XML document used to query and otherwise manipulate the CIM information base. It also specifies the format of the XML response to these requests (i.e. the returned string).</technologyDescription>

</BBContract>

5 BUSINESS PROCESS DRIVEN SYSTEM DEVELOPMENT GUIDELINE

5.1 Introduction and Objective of Guideline

This chapter presents the guideline for developing telecommunication service management system using a Business Process driven approach. The chapter first provides an overview of the development guideline and then presents the guideline specification in both text and UML. The guideline also illustrates the artefacts developed during the guideline to explain each workflow and illustrate an example outcome of each workflow. The guideline was applied and evaluated in several system development experiments as part of the FORM project. The evaluation of this guideline is presented in chapter 6.

Objectives of Business Process (BP) Development Guideline

The key objectives of the BP Development Guideline are to provide clear guidance as to how the development workflow is to be pursued and the design artefacts to be constructed to develop business process driven systems. In particular this involves defining workflows and artefacts to:

1. Support the modelling of Business and system process

2. Support the refinement and mapping of business and system activities onto Building Block Contract definition and associated information flows

3. Support the mapping of integrated (Business process driven) system designs onto existing BB and mapping of information flows between BBs that support the business systems design. It also involves identifying where missing functionality

is not supported by the BBs and support the development of such object and information modelling.

The objective of the guideline is to focus on the definition of design activities and workflows for modelling the business process system and assist the business process's realisation as an integrated set of BBs. It is beyond the scope of the guideline to provide detailed workflows for programming or coding.

5.2 Overview of Business Process Driven System Development Guideline

The guideline employs a business process driven approach to management system construction from re-usable BBs by explicitly modelling the required system processes and their constituent system activities. The guideline uses these system activities to determine the BB Contracts needed to implement these processes. Typically, this guideline is intended to be used by system integrators and service providers where they need to implement managed solutions using existing BB and BB Contracts offered within a service framework. Therefore an assumption of the guideline is that it is applied with a collection of BB and BB Contracts. The starting point of the guideline is the use of a reference architecture which describes the context over which the BBs and BB Contracts can be used. The guideline allows for the customisation & refinement of this reference architecture during the system modelling process.

The Guideline itself is divided into eight process workflows. Each workflow has a specific objective and produces or refines model(s) or artefacts. The workflows iterate the classic development activities of business modelling, requirements capture and management, system analysis and design modelling, implementation and testing [Fowler 97] but are focused on the construction of the system by using existing BBs and BB Contracts, and follow a business process driven approach.

The guideline specifies a mapping of the system activities to Building Block Contracts. This mapping is used to support the reuse of existing Building Block Contracts in implementing the management system processes. This mapping is at the heart of the reuse of Building Block Contracts in the implementation of management processes. A second part of the mapping of management activities to Building Block Contracts is the reconciliation of External Information Model(s) of the BB Contracts to the information flows in the system processes.

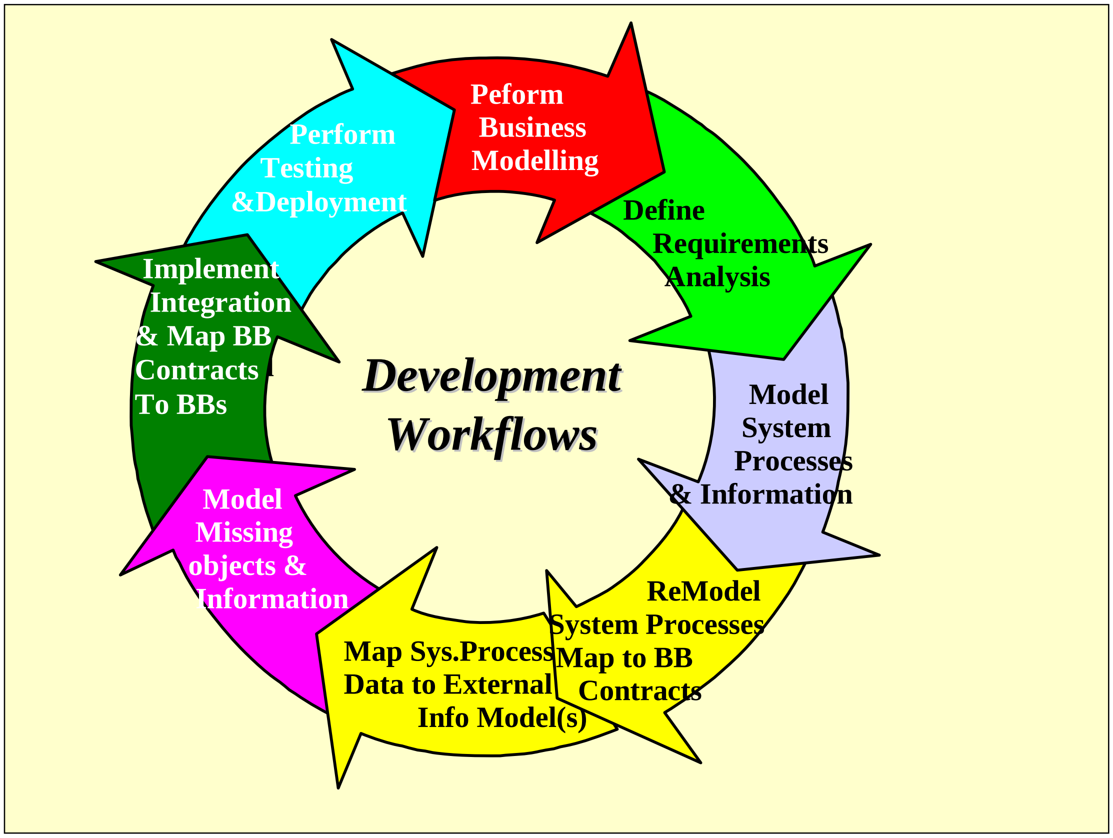


Figure 5.1 Overview of Process Driven System Development Guideline

Figure 5.1 identifies the principal workflows involved in the Guideline. These involve:

(i) *Performing Business Modelling:* This workflow facilitates the definition of Business Roles, business use cases and organisational units. The key results of this workflow are the development of Business Use Case Model(s), Business Model (representing the business roles and organizational units), and a refinement of the Reference Model.

(ii) *Define Requirement Analysis:* This workflow facilitates the identification of candidate behaviour of the management (business) processes, software requirement specifications and supplementary specification. The key result of this workflow is the system use cases and supplementary use case specifications.

(iii) *Perform System Process and System Information* Modelling: In this workflow the required system process(s) are represented as system activity diagrams. Thus, this workflow facilitates the modelling of system activities, their control flow and their information flows. The key results of this workflow are system activity diagram(s) representing the system processes to be implemented.

(iv) *Re-Model System Processes and Map to Building Block Contracts:* This workflow allows the mapping of system activities (and information flows) to Building Block contracts. This is one of the most important workflows in the guideline. In this workflow, the system activities are decomposed or aggregated to match, as closely as possible, available Building Block Contract interface specifications. This involves matching the BB Contract interface function(s) as well as their information requirements. Where matching is possible, the system activities are annotated with the Building Block Contract, which support it. Where the matched Building Block Contract requires extra information, these extra information objects have to be included in the system process. Where matching is not possible, the system activities will be modeled as bespoke system objects, which require separate design and implementation.

The key result of this workflow is the system process modeled as activity diagrams, with (some of the) system activities annotated with the Building Block Contract associated with them. The information objects in the activities diagrams are a combination of information objects drawn from the Building Block Boundary Information models and Information Objects developed specially for the process (i.e. bespoke information objects).

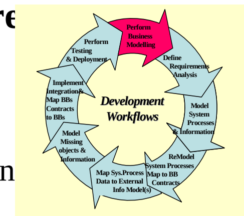
(v) *Model Missing Objects and Information Workflows:* This workflow supports the modelling of system objects and information, which is not supported by the chosen Building Block Contracts. This workflow concentrates on the bespoke development of management system functionality/objects, which have to be developed, as there is no appropriate Building Block Contracts to readily support it. The system objects are modeled as use cases, activity diagrams and class diagrams. Where a system activity involves the design of several system objects, they are grouped together in a subsystem package.

(vi) *Implement Building Block Integration:* This workflow facilitates the integration of the BBs so that the intended system processes are executable. This step may be automated if the design tool (used for modelling the system activity diagrams) is capable of generating an XMI description of the system activity model. Such an XMI description can be used to determine the control flow, data flow and appropriate BB Contract invocations. Depending on the integration implementation approach, such an XMI description could be used to populate a Business (Logic) Object capable of making the necessary decisions and invocations on the appropriate Building Contracts (interfaces), or could be used to populate a workflow specification for a workflow execution environment. The key result of this workflow is the implementation of the control and information flow for the business process.

(vii) *Map Building Block Contracts to Building Blocks and deploy BBs:* This workflow facilitates the selection of BBs to be deployed in the system. This workflow can also involve identification and placement of technology and data gateways where BBs are implemented using different technologies (be it Business (Logic) object or workflow integration engine). This would be needed if the BB Contracts were technology neutral (i.e. the interface description of the BB Contract was not described in a particular distributed implementation technology). In this way the technological or information representation heterogeneity of the BBs can be hidden from the integration business logic. Where the BB Contracts are specified using technology specific interfaces, there is no need for these gateways as the Business (Logic) Object or workflow engine would be generated with the required technology specific invocations.

(viii) *Perform testing and Deployment:* This workflow defines and executes the testing necessary for the management process execution. This involves generating test plans and execution of those plans.

5.3 Perform Business Modelling and Reference Architecture Refinement Workflow



This process workflow facilitates the definition of business model(s) based upon management business processes. In particular the workflow defines the Business Roles, Business Use Cases, Business Processes and Business resources/information entities with which the Actors interact. There are two aspects from which the Business Modelling is performed. The first is the External View of the Business, the second being the internal view of the system. To model (telecommunication service management) Business Processes the following models/diagrams are specified

(iv) Business Use Case Diagram(s): depicting business roles (workers and/or organisations), use case name (external view)

(v) Use Case Realisation which models the business workers and entities/resources needed to carryout the use case (internal view)

(vi) Activity Diagram depicting the activities involved in carrying out the use case (internal view).

The workflow can be summarised as Figure 5.2

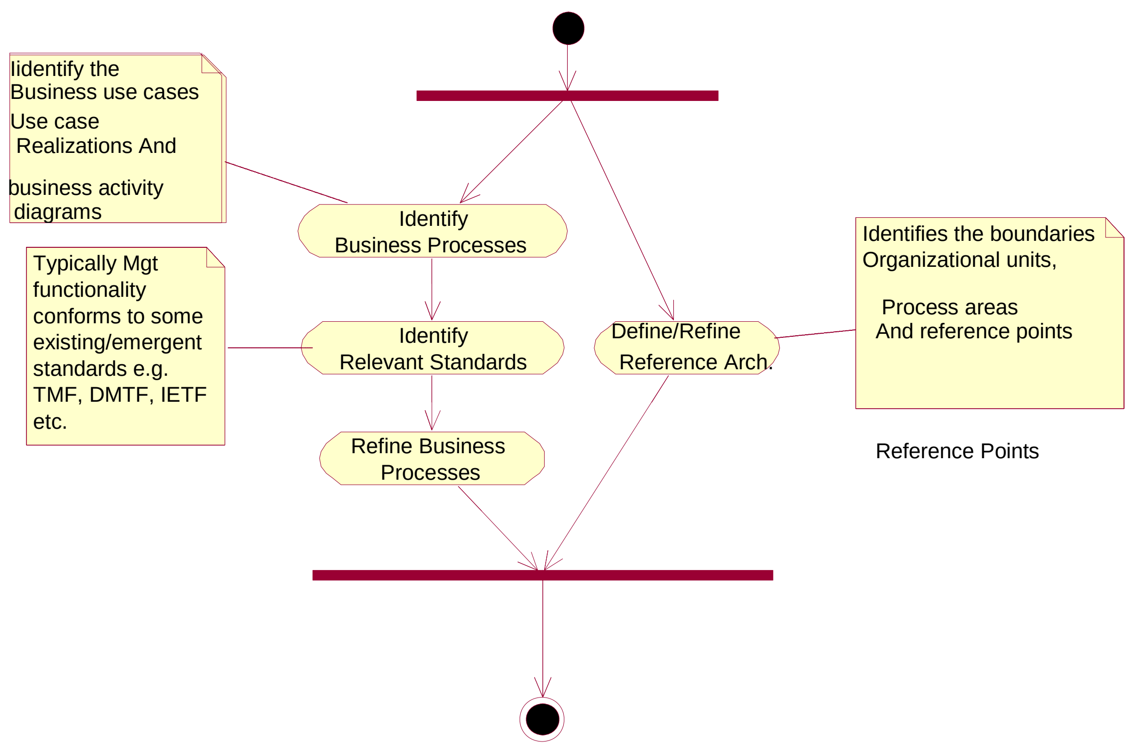


Figure 5.2 Perform Business Modelling and Reference Architecture Refinement Workflow

In developing the Business Model, a customisation reference architecture can be made[1]. The Reference Architecture identifies reference points between organisational boundaries, the placement of desired management processes within these boundaries and the relationships (potential interaction) between these process areas across organisational boundaries. The Business Modelling workflow therefore may involve either extending or simplifying the reference architecture to suit its business circumstances and context of the management processes to be developed.

[1] Where the Building Block Reference guidelines has been followed, there already exists a reference architecture.

Several external influences will very probably influence both the refinements of the reference model and the business model produced. These include standardization in the management process and technology areas e.g. TeleManagement Forum, DMTF as well as standards specific to the management process domain e.g. IETF/DMTF Quality of Service standards, IPDR accounting standards. It is important to identify the relevant standardization influences as this can have specific effects on system activities, use cases and information models.

Key artefacts produced by this process workflow include:

- A textual description of the scope of the envisaged management processes and the business organisation and roles involved in these processes.
- Business Use Case Model(s),
- Business Model (representing the business roles and organizational units), and
- A refinement of the Reference Model for the ODF framework (i.e. a specialization of the ODF reference model indicating the management processes and reference points to be used).

5.3.1 EXAMPLE: Assuring a Web based Educational Information service

Suppose a Customer Organisation wishes to subscribe to a Web based Application Service, offered by an Application Service Provider (ASP). In this case suppose that the ASP is in fact an online educational service provider offering a Web based education service (e.g. hypermedia delivered web course). The Customer Organisation may wish some guarantees regarding the quality of service for the web based services and the network performance upon which these services are offered. Such Quality of Service management is offered by an Inter Enterprise Service Provider (IESP). The network quality of service could be offered by a Virtual Private Network Service (VPNS) Provider. This IESP establishes relationships with the ASP and the Customer and Internet Service Providers that link the ASP and Customer. The IESP agrees a guaranteed quality of the VPN with the VPNS provider. It may be that this VPNS provider, in turn utilizes a Guaranteed Quality IP provider to ensure the quality of service over the IP connections between the end customer and the application service provider.

Typically the business modelling work would first identify the important business use case (depicted in figure 5.3).

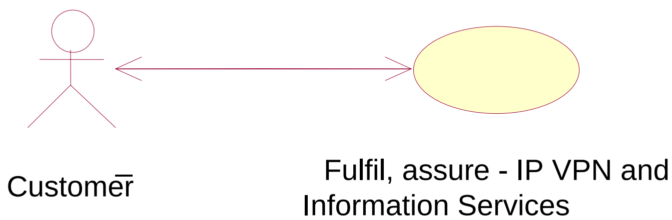


Figure 5.3: Fulfil & Assure the Educational Information Service and VPN.

The Business Model would therefore look like:

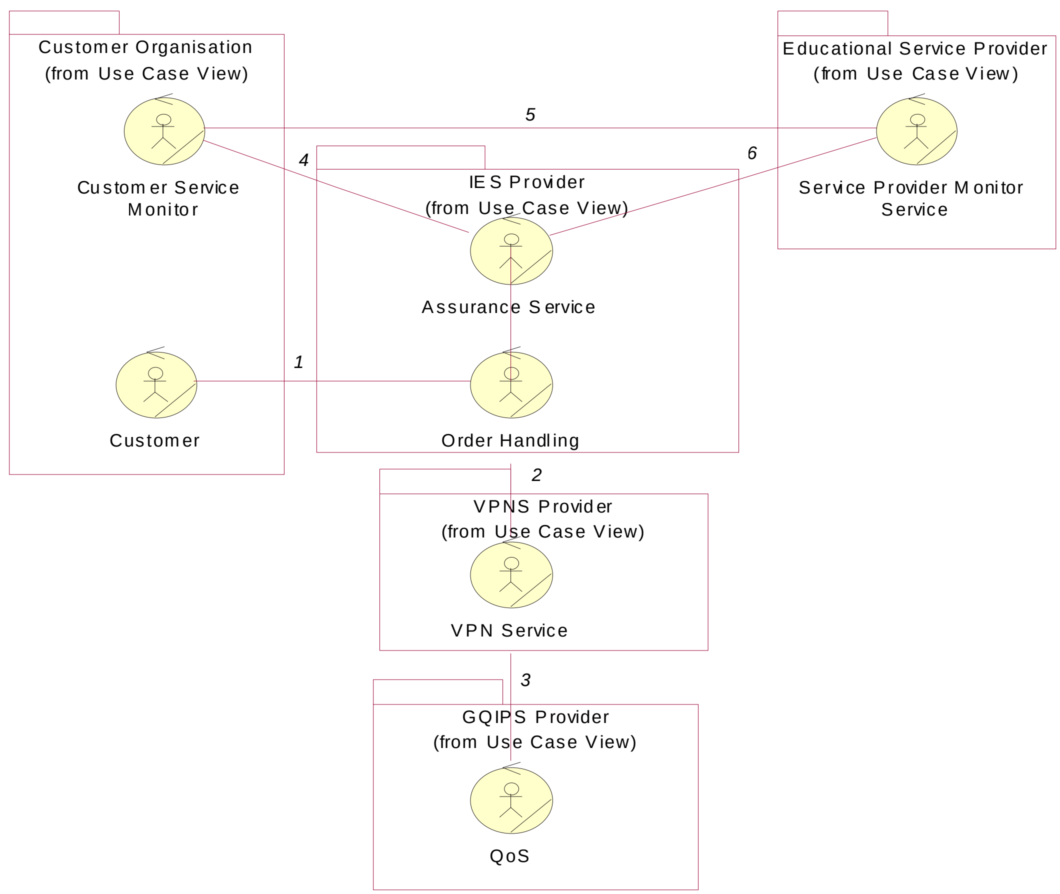


Figure 5.4 Business Model for Assurance Management Processes

The Business model, in Figure 5.4., depicts the organizations, the business roles and the relationships between the organizations. Each folder represents one of the organisations in the business model, namely: Customer Organisation, Educational Service Provider, Inter Enterprise Service (IES) Provider, Virtual Private Network Service (VPNS) Provider and Guaranteed Internet Service Provider (GQIPS). The GQIPS provider is responsible for providing the guaranteed bandwidth IP based service between IP Service providers. The VPNS provider is responsible for ensuring the appearance of a single network (virtual network) connecting the IES Provider and its customers. The IES

Provider offers a virtual end-to-end *assured* connection service between its customers (in this case the Educational Service Provider and the Customer Organisation of the Educational Service Provider). Thus the IES is an independent 3rd party who is responsible for providing added value management services e.g. (performance) assurance, service level agreement monitoring etc. between the Educational Service Provider and any of the Educational Service Providers customers.

Each organisation has at least one role represented. For the customer organisation, two roles are represented, namely the consumer and the customer service monitor. The consumer represents the actual consumer of the educational services and the customer service monitor is responsible for monitoring the service as it is delivered to the Customer organisation. Also modelled in this workflow is the customized version of the Reference Architecture as depicted in Figure 5.5. This customized reference architecture identifies the process areas within the organizations and the inter-organisational reference points operating between these organizations.

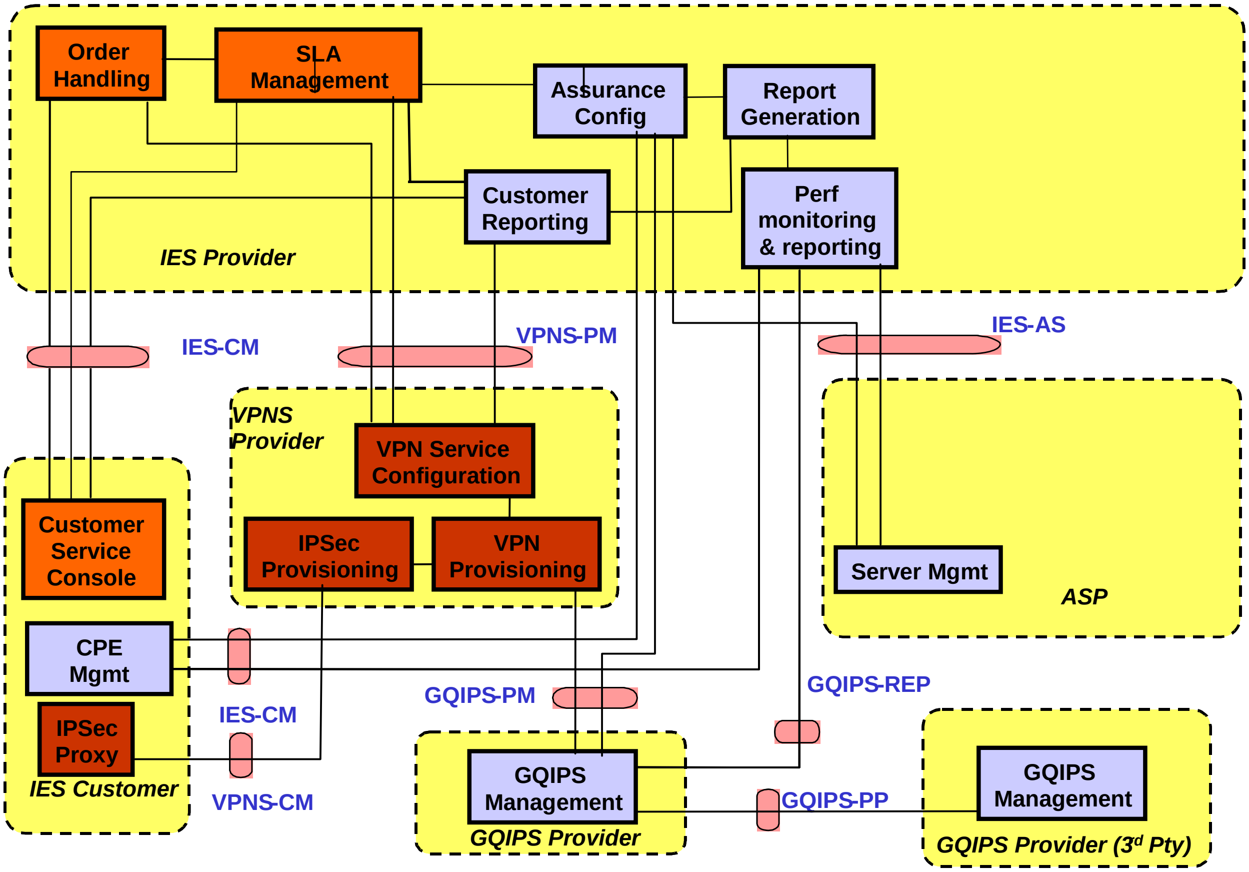
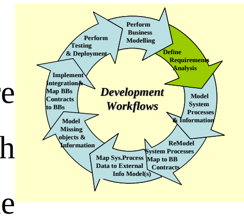


Figure 5.5 Reference Architecture for Fulfilment & Assurance Processes.

This reference architecture identifies the names of the processes relevant to the business use cases, the inter and intra process interactions within and between the organizations[1] identified in the business model and the reference points through which inter-organisational processes interact.

[1] Note in the reference architecture, the generic acronyms are given for the organisations. For example the reference architecture depicts the Application Service Provider (ASP), in the example this is the Educatioanl Service Provider.

5.4 Define Requirements Analysis Workflow



In order to identify candidate behaviour of the management processes, software requirement specifications and supplementary specifications are developed. Such requirements may be based on a market analysis of customers with regard to the functional areas. Other requirements may be gleaned from standards bodies and published requirement specifications. The use cases and the functionality identified within them, is at the ‘system modelling’ level (rather than the business modelling level). What this means is that the requirements modelling work is trying to identify, functionality to be supported by computer systems rather than higher-level business activities. The define requirements workflow is presented in Figure 5.6.

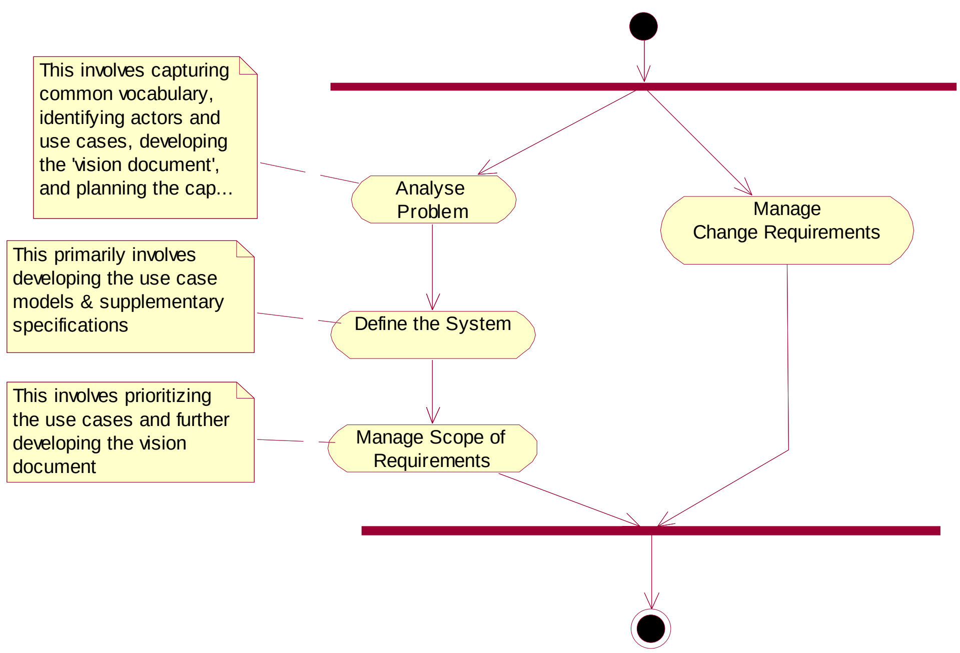


Figure 5.6: Define Requirements Analysis Model Workflow

During these development activities, Use Case Models are developed which describe the desired (system) behaviour of the envisaged management systems. Use cases at the boundaries in each of the process areas are developed. These use cases identify both the actors (roles), which would make use of the management services, and a specification of each of these management services as a use case. The use cases consist of Use Case Model diagrams, supplementary specifications and activity diagrams representing the control flow between the activities.

Thus the key artefacts produced by this workflow are:

- System use cases and supplementary specifications

5.4.1 Example Use Case Model for Fulfilment-Assurance

Taking the same example as before, the use case model for the IES provider could involve the initialization of the Service Level Agreement (SLA) between the end customer and the IES provider, the conclusion of the SLA negotiation and, as a consequence of this SLA negotiation conclusion, the instigation of the configuration of the assurance service, VPN and GQIPS. The folder notation in Figure 5.7 indicates that the use cases are contained with various roles defined earlier in the business model.

Although not shown here, these use cases diagrams would also have use case requirements specifications developed for them.

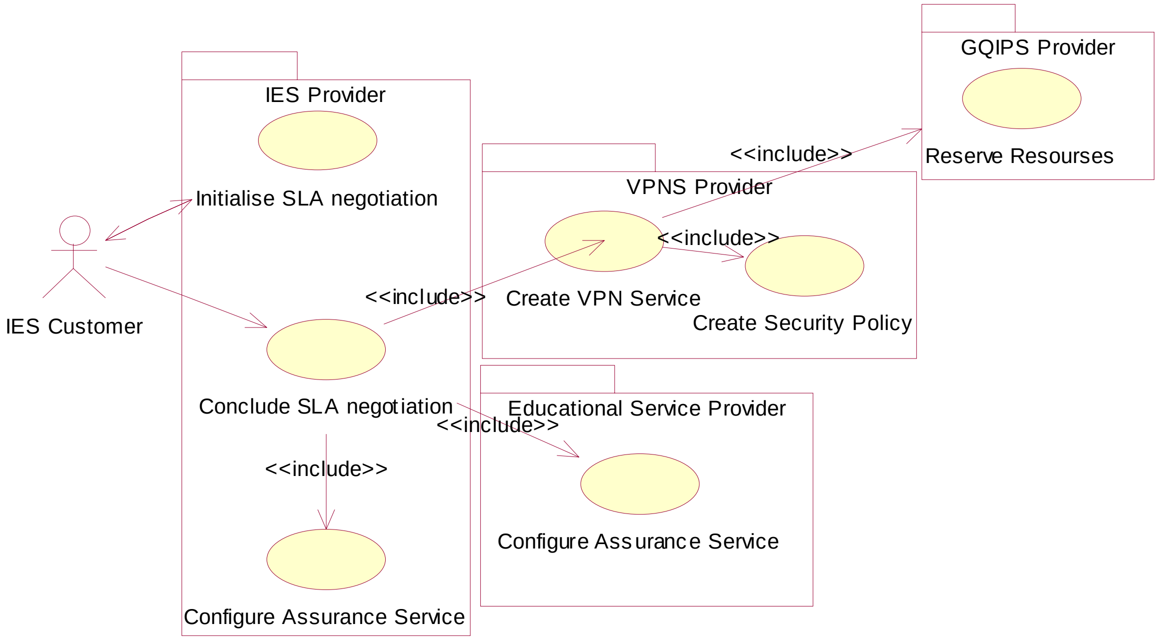
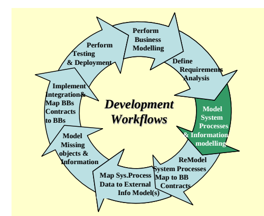


Figure 5.7: Example Use Model for assurance configuration

5.5 Perform System Process & System Information Modelling Workflow



This workflow involves the modelling of system activities, their control flow and their information flows, which occur in the management system processes to be developed. These management processes are identified from the use cases defined in the previous workflow.

The models produced by the workflow are:

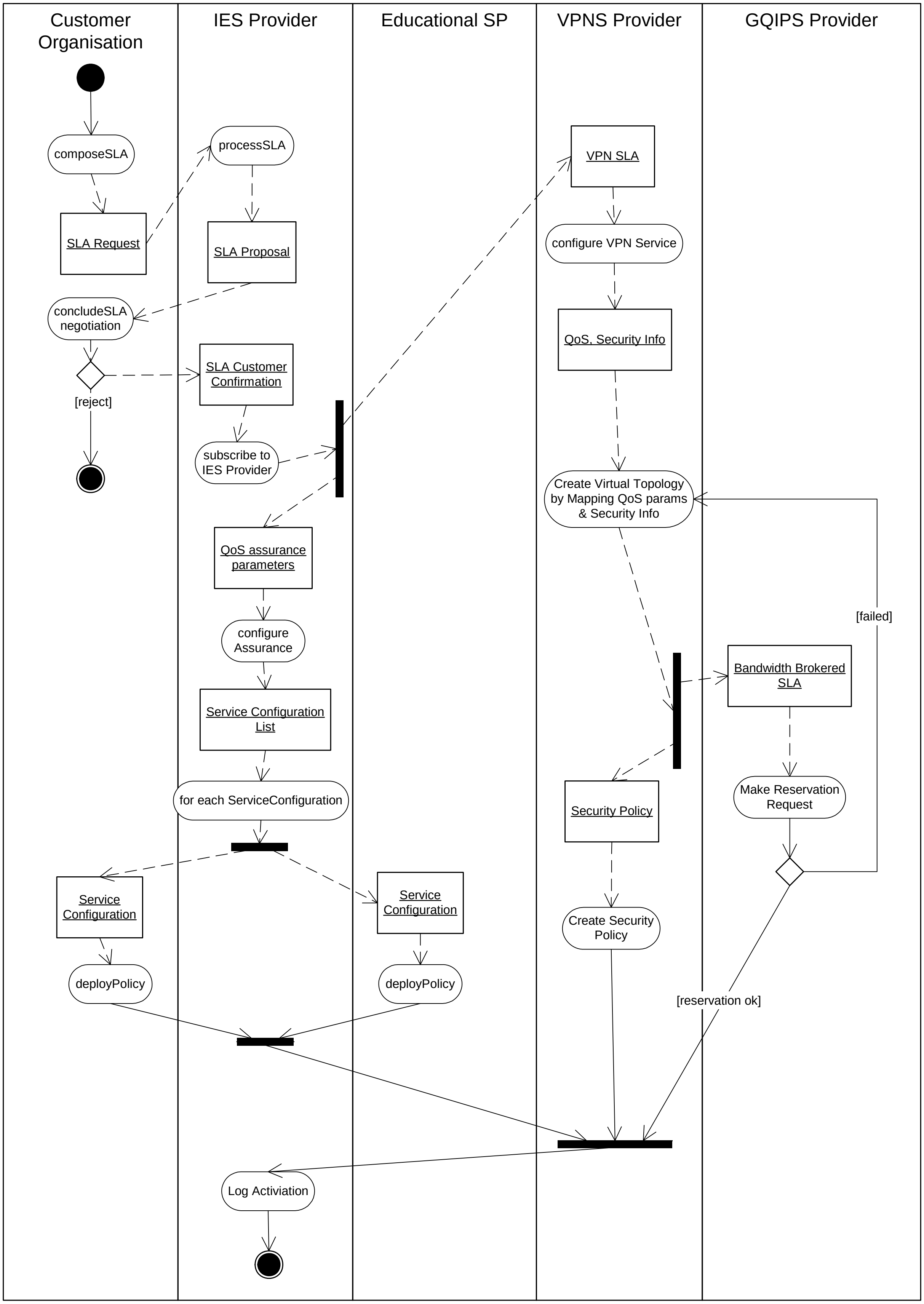
- Management System Process activities and Control Flows captured as Activity diagrams (graphs). In UML v1.4, these activity diagrams can also represent information objects, which are passed between the system activities.
- Optionally, a collaboration diagram with analysis objects representing system activities (as defined in the system activity diagrams) and objects from the class diagrams representing data flows between these analysis objects. Some developers prefer this representation; however, it does not make explicit the control flow decisions, which govern the interaction between the system activities.

5.5.1 Example: System Process for Fulfilment/Assurance System for the Educational Service Provider

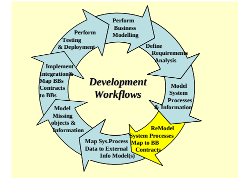
If we continue the Fulfilment-Assurance example from previously, this workflow would indicate that the next development task is to develop a system activity model for Fulfilment-Assurance. This system activity diagram needs to show the system activities, control flow, information objects, branching, iteration and synchronization points. The swimlanes are used to represent the different roles in the management processes. Flows across the swimlanes are used to indicate control or data flows across organizational (role) boundaries. The level of granularity of the system activities are at a relatively high level of abstraction and the information flows are based on information either already specified in the External Information Model are based on some appropriate standard information model e.g. DMTF, TeleManagement Forum. The modelling of the system activity diagram(s) requires several iterations to ensure consistency with appropriate information models and activity granularity etc.

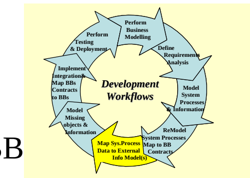
An example business process developed for the negotiation of a SLA for service assurance is presented in Figure 5.8. Here the business process spans all five organisations (represented by swimlanes) and depicts the activities to request and set up a managed (educational) service across a Virtual Private Network Service with guaranteed bandwidth IP connection. The business process defines activities performed by the Customer Organisation, Inter Enterprise Service (IES) Provider, Educational Service Provider, Virtual Private Network Service Provider (VPNS) and Guaranteed Quality Internet Service Provider (GQIPS). The customer provider first composes a SLA request which the IES service provider will attempt to setup and manage. This SLA is negotiated between the IES Provider and the Customer Provider. Once this SLA is agreed with the IES Service provider, the necessary Virtual Private Network (VPN) Service Level Agreement is requested. The VPN Service Provider is responsible for configuring the necessary VPN service between the Customer Organisation and the Educational Service Provider, and for defining the necessary security policies and quality of service parameters. The VPN provider makes the necessary requests (and SLA) for the IP network from the GQIP Service Provider.

In parallel to the VPNS Provider activities, the IES Provider configures the necessary assurance management functions so that the educational service can be monitored (and assured) between the customer organisation and the Educational Service Provider. The necessary service management policies are deployed on the Customer Organisation site and the Educational Service Provider site. These tasks having completed, the IES Provider logs the activation of the assurance of the educational service and completes the business process.


Customer Organisation
IES Provider
Educational SP
VPNS Provider
GQIPS Provider
composeSLA
processSLA
VPN SLA
SLA Request
SLA Proposal
configure VPN Service
concludeSLA negotiation
QoS, Security Info
SLA Customer Confirmation
[reject]
subscribe to IES Provider
Create Virtual Topology by Mapping QoS params & Security Info
QoS assurance parameters
[failed]
configure Assurance
Bandwidth Brokered SLA
Service Configuration List
Make Reservation Request
for each ServiceConfiguration
Security Policy
Service Configuration
Service Configuration
Create Security Policy
deployPolicy
deployPolicy
[reservation ok]
Log Activiation

5.6 Re-model System Processes and Map to Building Block Contracts workflow; Map System Process data to Boundary Information Model(s)





This workflow uses the activity definitions and diagrams to identify candidate BB Contracts, which offer equivalent/appropriate behaviours. However it is unlikely that exact matches of behaviour and the appropriate level of granularity will be readily available. Therefore a re-organisation of the system development processes is necessary to decompose system activities to the granularity that matches some of the BB Contract behaviours. It is unlikely that all activities of a system process will be performed by BB Contracts. These missing functionalities are identified for later development effort in the guideline.

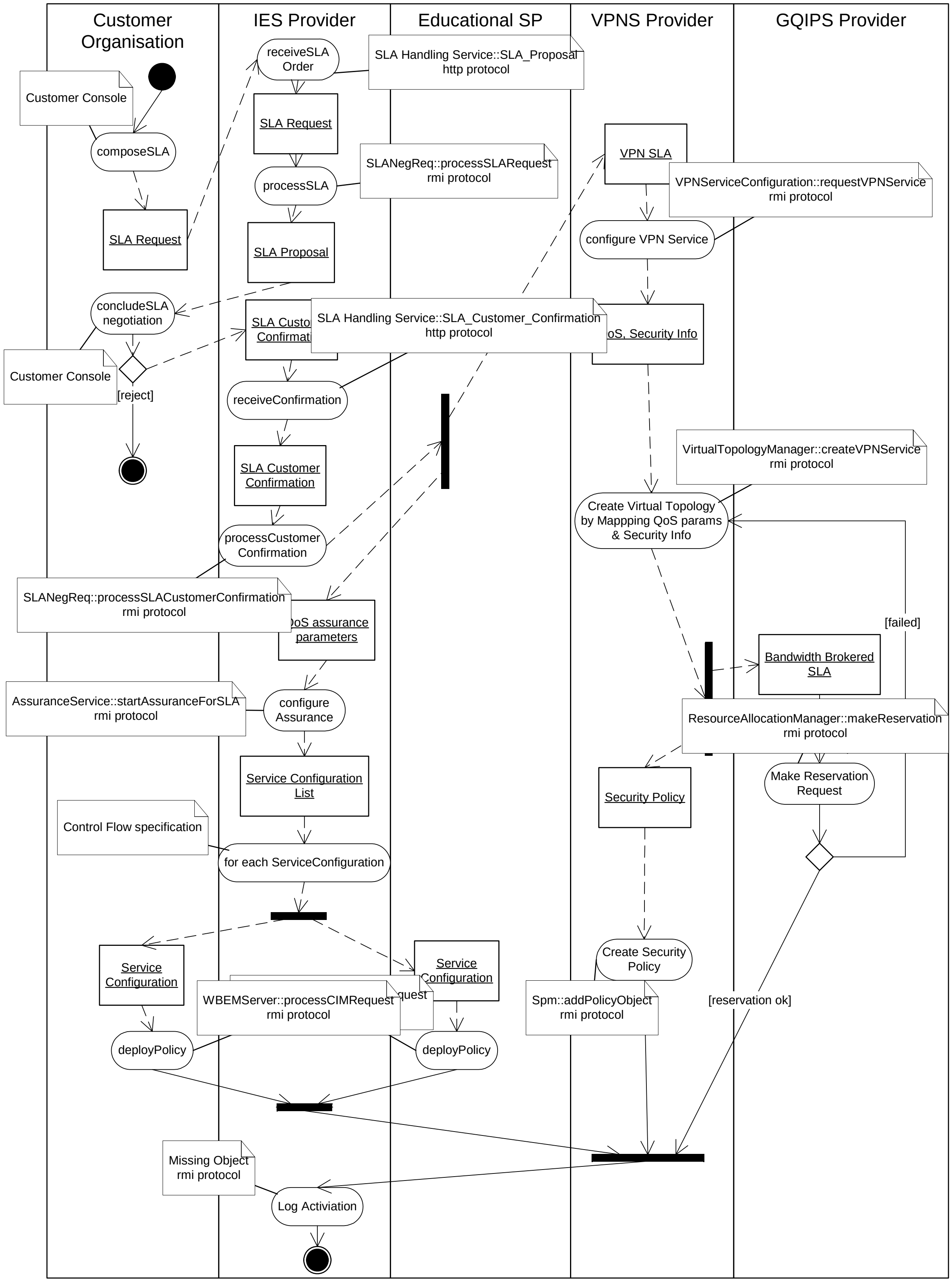
However, behavioural re-arrangement is not the only cause for reorganising the System Processes. Because of the likely mismatch between the chosen BB contracts' Boundary Information Models and the information flows in the system process(es), extra activities which perform data retrieval, transformation or generation may also be required. This will make the reorganised system process control flow consistent with the information flow within the system processes. This remodelling work will need to be iterated a number of times until a satisfactory balance between reuse of BB Contracts and development of new software (i.e. missing functionality) is achieved.

The result of this workflow is (Figure 5.9):

- A new set of system activity diagrams, in which each activity is noted as being supported by a (named) BB Contract or is noted as not supported.
- Again, optionally, collaboration diagrams indicating information flows between the analysis objects each of which represents a BB Contracts in the catalogue, or a new object which needs to be implemented to support the management processes can also be modeled.

5.6.1 Example: System Activity Diagram with Building Block Contract Annotation

The example presented in Figure 5.9 illustrates the business process as previously depicted in Figure 5.8, but this time the activities are annotated with the building block contracts which can implement the activities specified in the business process. These building block contracts provide the explicit mapping of the business process to sequencing of building block contract invocations needed to implement the business process.


Customer Organisation
IES Provider
Educational SP
VPNS Provider
GQIPS Provider
Customer Console
composeSLA
SLA Request
concludeSLA negotiation
Customer Console
[reject]
receiveSLA Order
SLA Handling Service::SLA_Proposal http protocol
SLA Request
processSLA
SLANegReq::processSLARequest rmi protocol
SLA Proposal
SLA Handling Service::SLA_Customer_Confirmation http protocol
receiveConfirmation
SLA Customer Confirmation
processCustomer Confirmation
SLANegReq::processSLACustomerConfirmation rmi protocol
QoS assurance parameters
AssuranceService::startAssuranceForSLA rmi protocol
configure Assurance
Service Configuration List
Control Flow specification
for each ServiceConfiguration
Service Configuration
Service Configuration
WBEMServer::processCIMRequest rmi protocol
deployPolicy
deployPolicy
Missing Object rmi protocol
Log Activiation
VPN SLA
VPNServiceConfiguration::requestVPNService rmi protocol
configure VPN Service
QoS, Security Info
VirtualTopologyManager::createVPNService rmi protocol
Create Virtual Topology by Mappping QoS params & Security Info
[failed]
Bandwidth Brokered SLA
ResourceAllocationManager::makeReservation rmi protocol
Security Policy
Make Reservation Request
Create Security Policy
Spm::addPolicyObject rmi protocol
[reservation ok]

5.6.2 Example External Information Model for Fulfilment-Assurance

In the example, several information objects need to be passed between the BB Contracts. The workflow identifies these information objects and those that are passed between the BB Contracts that are involved in the process. The first four information objects passed between the BB Contracts are SLA_Request, SLA Proposal, SLA_Customer_Conformation and Assurance_SLA. These obejcts are defined in the External Information Model. Figure 5.10 depicts the Assurance SLA part of the external information model.

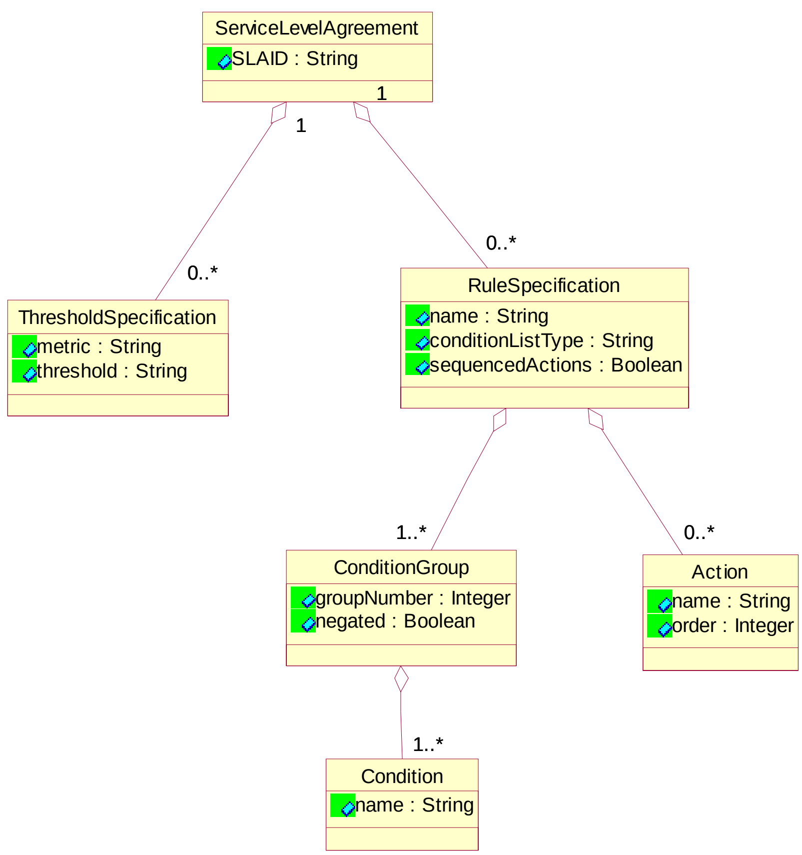
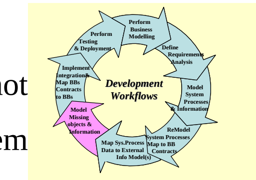


Figure 5.10 SLA Assurance (part of External Information Model for Fulfilment-Assurance Business process)

5.7 Model Missing Objects and Information Workflow



This workflow supports the modelling of the objects and information which are not supported by BB Contracts. These analysis objects are first identified in the system activity diagram (representing the system processes). Some of these analysis objects may be coalesced into one or more analysis objects and, where they interact directly with each other, packaged into a subsystem. The workflow follows standard software analysis and design activities for the modelling of the classes and the software design to support these objects is then performed.

The missing objects development activity can be done ad-hoc as the classes developed are not intended to be reused across business processes. Only after time, if there are several similar classes being developed or classes which support the processing/manipulation or management of the same resource, that a Building Block design could be considered. At this point the BB guideline can be used to refactor the classes and existing designs.

5.7.1 Example Missing Object and Information Workflow

In this example, suppose a Log Activation activity requires new system functionality to be modelled and implemented. The system use case is identified and shown below (Figure 5.11).

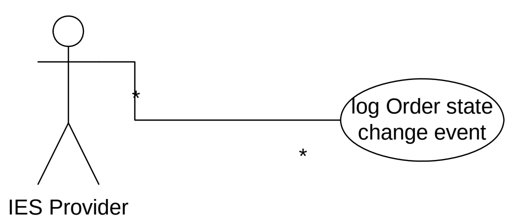
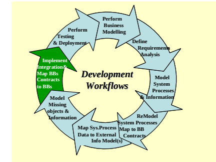


Figure 5-11 System Use Case covering the Log Activation functionality

The development of this use case into further system models would be developed further using standard RUP class development approaches. The final design could be implemented as a Class library (or RUP package).

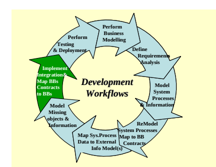
5.8 Implement Building Block Integration



The objective of this workflow is to facilitate the implementation of the necessary ‘application integration logic’. Such integration implements the control and data flows represented in the system activity diagrams. The integration allows the invocation of BB Contract interfaces and the bespoke object interfaces (developed in the previous workflow).

This integration logic can be hand crafted in a Business (logic) object. In some circumstance, where the tool used to model the system activity diagrams supports some standard process description languages e.g. XMI, this business object can be (semi) automatically generated. If a workflow engine integration approach is used to integrate the BB Contract Interface invocations, then the ‘process rules and data flow rules’ for the process automation engine can be generated. The Guideline does not dictate the building block integration technology. However in the example used and in the validation trials, a workflow engine was used to provide the integration.

5.9 Map Building Blocks Contracts to Building Blocks & deploy BBs and Business (Logic) Object(s)

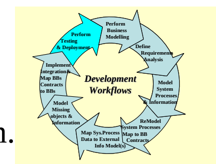


This workflow facilitates mapping of BB Contract onto BBs. The BBs may be technology dependent and therefore the necessity of using technology or protocol gateways may be required to allow BBs to interoperate.

The result of this workflow is a full specification of the management system components as it cooperates to support the management processes. The artefacts produced by this workflow will include:

- Activity diagrams where each activity is a component in the management system. The activity diagram will show the control flow of the system management process.
- Class/object Model which models the information passed between the system components
- Collaboration Diagram with each component in the system modelled including the passing of information between the components.

5.10 Perform Testing and Deployment Workflow



This workflow defines and executes the testing necessary for the management system. The workflow involves the design and implementation of test plans as well as their evaluations. These test plans are based on the system use cases defined earlier in the guideline. The workflow also involves planning and assembly of packages so the system can be deployed onto the target environment. This involves the planning and execution of the deployment of the BBs and other object implementations to ensure the proper execution of the system.

Appendix 2 contains a presentation of an example case studies which used the BB development guideline for designing BBs and BB Contracts in the area of Billing and Accounting of services. This appendix also presents an example Provisioning system (for VPN) which was developed using the Business Process driven system development guideline. These case studies were presented as part of a tutorial at NOMS 2002 [Wade 2002a].

5.11 Summation

Chapters four and five have presented the BB Development Guideline and the Business Process Driven System Development Guideline. Full descriptions of each of the guidelines were published as part of the FORM evaluation [Wade 2002a]. Full description of the BBs and the Business Process Driven Systems which were implemented using MODD were also published [FORM 2002b]. This includes a description of the 'Catalog' of Building Block Contract developed using the MODD. Finally a technical report describing the evaluation of the BBs, Building Block Contracts and the Business Process driven systems was also published [FORM 2002c]. Chapter 6 presents an evaluation of the MODD guidelines and provides an analysis as to it relationship with key research identified in earlier chapters.

6 TRIALS AND EVALUATION OF DEVELOPMENT METHODOLOGY

6.1 Introduction

This chapter provides an evaluation of the two MODD development guidelines. In order to evaluate the guidelines, they were applied within a large European funded research project called FORM (IST IST-1999-10357). The overall goal of the FORM project was to develop a framework of reusable telecommunications management services and validate this framework by rapidly constructing telecommunication service management systems which were implemented by these framework services. In order to investigate the usefulness and ease of use of MODD, the developers who had applied the guidelines in the development of the FORM services and systems, were surveyed via questionnaire. Section 6.1 introduces the evaluation process which was conducted as part of the FORM project. This section identifies the evaluation approach and outlines the objectives of the evaluation. Section 6.2 describes the objectives and evaluation results for the usage of the Building Block Development Guideline. This evaluation focused on analysing the developers experience in applying the guideline and evaluating the guideline usability. This evaluation is based on questionnaire survey results and follow-up interviews with the component and service developers involved in the Building Block and BB Contract development in FORM.

Section 6.3 focuses on the evaluation of the business process system development guideline which was conducted as a separate trial. Again the evaluation focuses on analysing the developers experience in applying the systems development guideline and its usability in realising management solutions. The section describes the evaluation results which are based on questionnaires completed by application developers involved in the FORM systems development.

Section 6.4 examines MODD in relation to other telecommunications management development processes and related research. More specifically it provides a comparison of aspects of MODD with methodologies prevalent today in telecommunications management and indicates MODDs uniqueness among them and the benefits derived from MODD.

6.2 FORM Project and Evaluation Approach

The FORM project consisted of eleven different organisations, representing most of the stakeholders roles in the telecommunication market (previously identified in Figure 2.1), namely telecommunications service providers, telecommunication component vendors, telecommunication systems developers and systems integrators, and organisations who were active in the standards fora. Each of the organisations was focused on providing different management functionality and intended using a variety middleware technologies (e.g. EJB, CORBA, XML/HTTP, SOAP, Workflow Execution engines etc.).

The objective of the FORM project was to develop an open framework consisting of many different reusable (telecommunication management) services. These framework services were focused on the management of business-to-business services over QoS enabled IP networks. The framework focused on two value chains namely the (telecommunications management) system provision chain and the (telecommunications management) service development chain. The system provisioning value chain can be considered as a collaboration of organizations to develop a set of management systems to provide quality of service based management of IP based networks and the applications running over them. Thus it sought to develop seamless service management of network services across organizational boundaries (i.e. develop seamless management services capable of managing applications or network services which operate across a chain of enterprises). The service development chain involved organisations who sought to rapidly develop reusable component software (and services) to satisfy rapidly changing telecommunication service management requirements. Thus this chain involved collaboration of organization wishing to populate the open framework with sets of reusable components and services.

A key element in realising this framework of services was the MODD development guidelines which directed both the development of telecommunication management services but also directed the development of management systems.

6.2.1 Evaluation Approach

The objective was to evaluate the usability and appropriateness of the two MODD guidelines in developing telecommunications management BBs, Building Block Contracts and Business Process based systems. The evaluation sought to examine the developers usage and experience at different stages in applying the guidelines.

The systems and services being developed were of contrasting requirements (and involved different international standards). The management areas for these services and systems included quality of service management, performance monitoring, service level agreement management, accounting and billing, and security (IPSec).

To perform the evaluations, two trials were held. The 1st trial was focused on the development of reusable BBs and building block contracts in the general areas of Fulfilment, Assurance and Billing. Developers had freedom to apply the most appropriate standards to their service (which typically involved the use of DMTF, TeleManagement Forum and IETF). Each of the services designed was individually trialled and evaluated against test cases for functional coverage and operational performance [FORM 2002c]. The 1st trial involved the testing and demonstration of each of the BBs and an evaluation of the BB development guideline was conducted by surveying via questionnaire and interview, the developers of the BBs and BB Contracts.

The second trial was held to evaluate and validate the business process driven system development guideline. In order to evaluate and validate this guideline, four different systems were developed using the FORM services (and BBs). These (inter-organisational) management systems were a Virtual Private Network (VPN) Provisioning system, a Quality of Service Management System, a Network and Applications Service Assurance (performance management) System and a Billing system. The 2nd trial involved the demonstration and testing of each of the systems and an evaluation of the BP development guideline. A full specification of each of the Trial systems was developed using MODD and was published as the FORM Project public technical report [FORM 2002b]. Based on these trial system designs, prototype systems were implemented and demonstrated at public events in Dublin (Ireland) and Copenhagen (Denmark). The methodology, trial system designs and prototype implementations were also evaluated by three independent experts in the area of telecommunications management as part of the EU IST Project Review[1].

The evaluation presented in this chapter focuses on the guideline evaluation. Therefore presented in this chapter is an analysis of the developers experience in applying the two guidelines. An evaluation of the actual Building Block implementations and Business Process system developed in FORM is outside the scope of this evaluation. However, such an evaluation was conducted by the development team and published in a two volume technical report [FORM 2002c]. The implementations were based on four test cases (inter organisational business processes) involving the quality of service management system, a network and services assurance system, Virtual Private Network Fulfilment system and a billing system [FORM 2002d].

6.3 Building Block Development Guideline Evaluation

This section first outlines the method of evaluation used to capture how the guideline was assessed. It then identifies the key objectives of the evaluation and summarises the its findings.

Evaluation Approach

The evaluation was primarily performed via questionnaire completed by the FORM developers when conducting the design and development of the 1st trial BBs and BB Contracts. The development team consisted of BB designers, developers, project

[1] The panel of experts included a representative from the TeleManagement Forum, a chief architect from a telecommunications development company and a senior researcher in a systems integration multi-national organisation.

managers and reviewers. Eleven FORM developers completed the questionnaires and provided feedback as to their usage and opinion of the guideline.

In order for MODD to be clearly understood, the FORM project held several workshops to explain the usage and application of the building block development guideline. Also, during the development of the BBs and Building Block Contracts, joint meetings were held. These provided both an opportunity to learn the development process and relevant notations as well as providing feedback as to the presentation of the guideline[1]. During the development of the BB, collaborative working sessions were hosted where clarification regarding the development process could be given and best practice in its application was illustrated.

The trial of the BBs and building block contracts consisted of both the testing of the services developed (against their original intended purpose as defined the the requirements and use cases). The second aspect of the trial focused on the developer experience in applying the guideline. This section describes the evaluation of the developers experience in developing BBs and BB Contracts.

BB Guideline Evaluation Objectives

The objective of the BB Development Guidelines evaluation was to ascertain the experience of the FORM developers in applying the guideline. In particular to investigate the clarity, complexity, usefulness and benefit of using the BB Development Guideline. To achieve this, a questionnaire was developed which focused on the experience of the developers as they worked through the various workflows and developed the design artefacts. The questionnaire-based evaluation of the BB Guidelines was conducted subsequent to the 1st FORM Trial. The purpose of the questionnaire was to

[1] It is important to realise that the development of the MODD guidelines were iterative. This means that as the developers reached different iterations and workflows in the guidelines, feedback was taken, and where appropriate, improvements to the guidelines were made. The evaluations presented in this chapter are based on the final guidelines (as presented in chapter 4 & 5)

(i) Determine the prior experience/knowledge of the developers with respect to UML and RUP, as well as the roles of the developers in the development process.

(ii) Determine the developers own experience of the usefulness and usability of the Development Guideline and the design artefacts produced by the guideline

(iii) Examine the ease of modelling experienced by the developers in applying the Development Guideline

(iv) Identify any difficulties in using the development workflows or developing their design artefacts.

Appendix 3 contains a copy of the BB development process evaluation form, which 11 FORM developers completed. Appendix 4 contains the full set of graphs indicating their responses to each question. The sections below provide a discussion of the results of the questionnaires and an indication of their findings.

6.3.1 Estimation of Prior Experience of UML and RUP

Firstly, the FORM developers were asked to indicate their prior expertise/familiarity with UML (question 1) and RUP (question 2). The horizontal axis in both Figure 6.1. and Figure 6.2, represents a five point scale between expert and someone who had never used UML or RUP before. The vertical axis indicates the number of developers who indicated their expertise in a particular category.

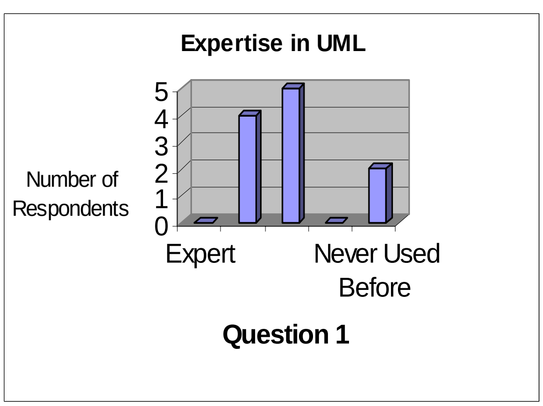


Figure 6.1 Prior UML Experience

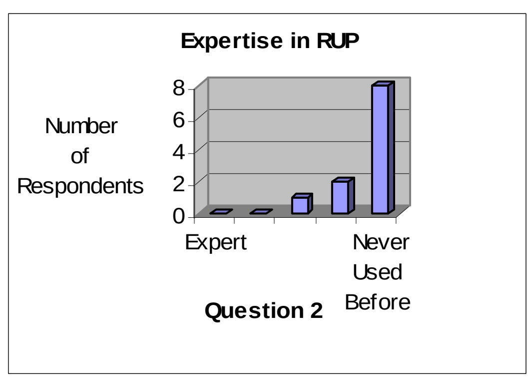


Figure 6.2 Prior RUP Experience

From the responses it is clear that although there was some expertise in UML, very few of the developers were familiar with the Rational Unified Process. The majority (80%) were familiar with UML diagrams. However, none considered themselves expert in its use and the largest group (45%) rated themselves as average in its use. Experience with RUP was much lower, with 72% of respondents indicating no prior experience of RUP and 18% indicating very poor prior knowledge.

This shows that the usage of the guideline did not benefit from, or rely on prior experience of RUP. It also indicates that the spread of experience in UML was quite wide.

6.3.2 Roles of the developers in the development process

The respondents to the questionnaire were drawn from different sections of the FORM 'partner organisations' development teams, however most respondents were principally involved with the development of BBs (rather than BB review or development management). Figure 6.3 indicates the percentage of respondents who were principally responsible for 'Development'(1), 'Implementation'(2), 'Management of Development'(3), 'Development Reviewer'(4), or 'Other'(5) respectively. The percentages indicate the relative number of guideline users with each responsibility. The overall number of responses (31) is greater than the actual number of respondents (11) as some respondents were involved in multiple roles. The most common 'double' role being Development and Implementation.

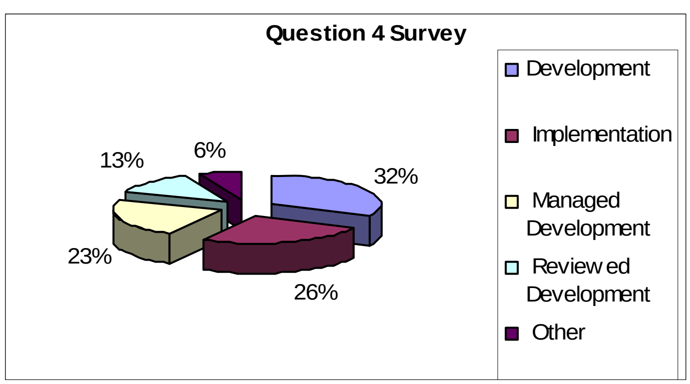
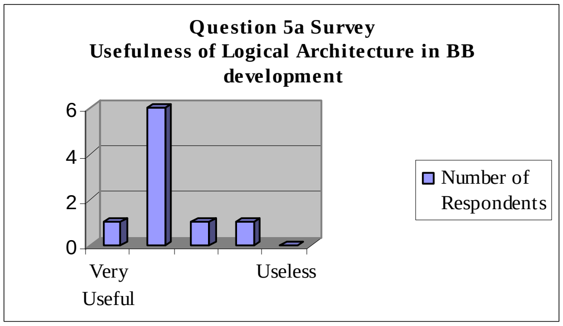


Figure 6.3 Roles of Respondents in Development Process

It is clear from Figure 6.3 that questionnaire respondents were principally involved in the design, development and management of the development of the FORM BBs. Only 19% of the roles performed by the developers were for review or ‘other’ purposes. This means that the responses of subsequent questions would be valid principally for the development, implementation and management of the development aspects of the BB Guideline. This is the target audience of the Guideline and means that the opinions expressed are relevant to the Guidelines evaluation.

6.3.3 Usefulness/Significance of Artefacts developed for describing a Building Block

One of the key criticisms of RUP and other model based development guidelines was that of excessive documentation and asset development, many of which were not that important in shaping the final implementation. Therefore a key question in the investigation of the guideline was “how useful were the artefacts produced during the application of the guideline for describing your BB to other developers?” This question (question 5 on the survey) was asked about (a) Logical Architecture Diagram; (b) Reference Points Supported by BB; (c) Contract(s) Supported by BB; (d) Contract descriptions supported by BB; (e) Use cases for BB (f) Collaboration diagrams to illustrate BB usage; (g) Contract interface specification; (h) Information Model; (i) Technology Description; and (j) Collaboration diagrams for other BB interactions (relied upon by the BB under development). Thus in question 5 (parts a to j) the respondents were asked to rate the usefulness of several models/artefacts used to describe their BBs e.g. Q5 part(a) asked the usefulness rating for the logical architecture (Figure 6.4), Q5 part(b) asked for their experience of the usefulness of the reference points (Figure 6.5). A sliding (5 point) likert scale was used, ranging from Very Useful to Useless.



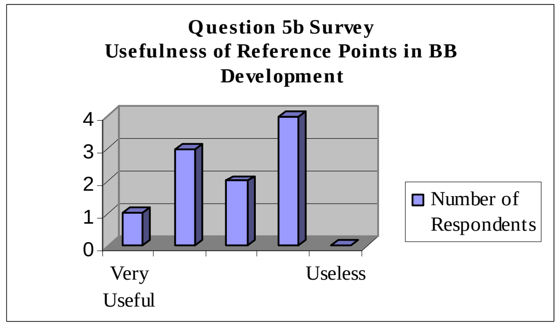


Figure 6.4 Usefulness of Logical Architecture

Figure 6.5 Usefulness of Reference Points

From the above responses and comments, it was clear that the logical (structural) architecture was very helpful to developers in associating the various FORM functionalities relative to one another. 78% of responses rated it as very useful or useful with no respondents rating it useless. However, at the level of BB, the developers did not necessarily see the benefit of reference points. 60% rated it as useful or very useful, but 40% saw little use in it. Subsequent interview revealed that the belief was that the BB was too low level to be concerned directly with reference points[1] on the organisational boundaries. As the BB were (reusable) in many different scenarios/circumstances, the Reference Points of the logical architecture were not that relevant except (perhaps) when implemented in a specific application.

When asked about the constituent artefacts/models used to describe BBs, the respondents gave very high appreciation to the usefulness of contract description (question 5d, Figure 6.7), use cases (question 5e, Figure 6.8), collaboration/sequence diagrams (question 5f, Figure 6.9), information model specification for BB (question 5h, Figure 6.10).

[1] Reference Points were used on the Reference Architecture to indicate points at which inter-organisational boundaries would be crossed. This is important when constructing inter-enterprise management systems as these interfaces tend to require more care, security and are more likely to be loosely coupled.

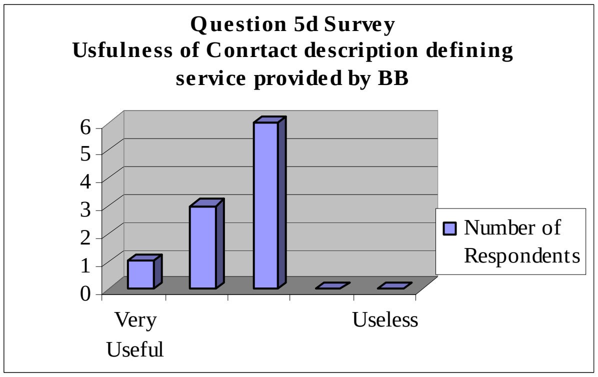


Figure 6.7 Usefulness of Contract Description

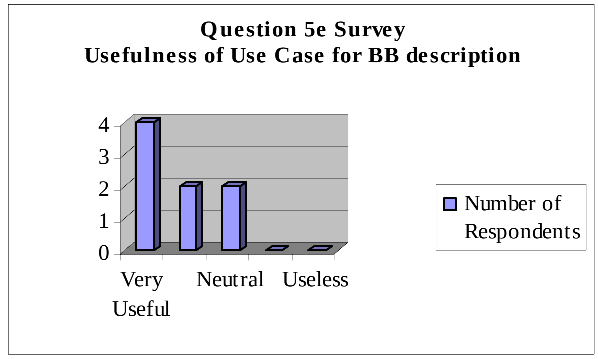
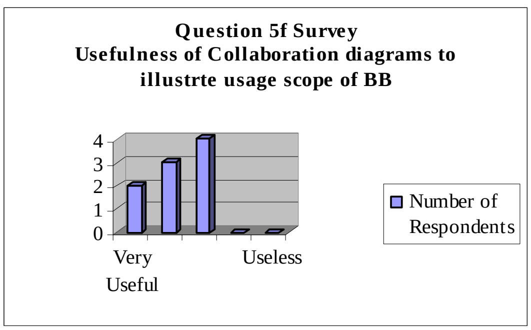


Figure 6.8 Usefulness of Use Case



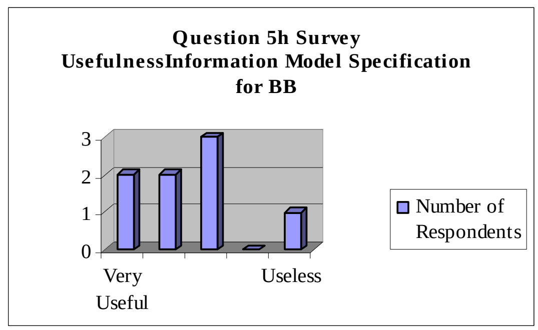
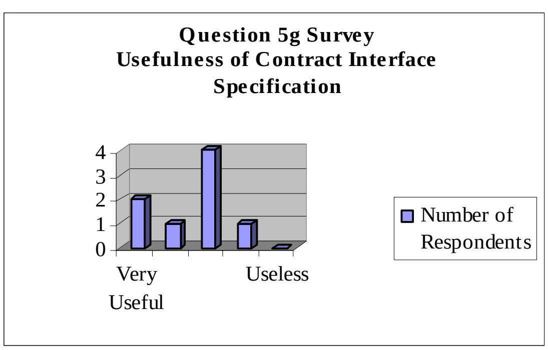


Figure 6.9 Usefulness of BB Collaboration Diagram

Figure 6.10 Usefulness of Info. Model Specification for BB

Respondents found the contract interface specification, technological description artefacts describing a BB slightly less useful (as depicted in Figures 6.11 and 6.12). However, these results still reflect a high average satisfaction for the BB description artefacts.



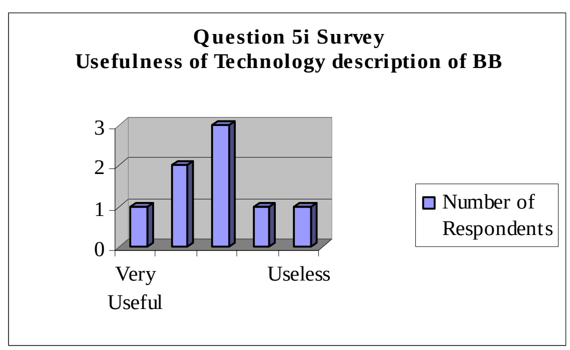


Figure 6.11 Usefulness of Contract Interface Specification

Figure 6.12 Usefulness of Technology Description

Overall, respondents seemed content with the level of BB representation. A full breakdown of all the results is presented in Appendix 4 of this document.

6.3.4 Business Modelling

When asked how relevant were the business models developed for FORM with regard to the actual BB development, most respondents indicated a weak relevance as depicted in Figure 6.13.

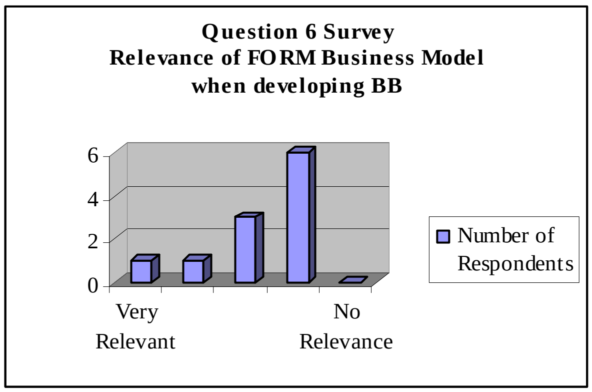
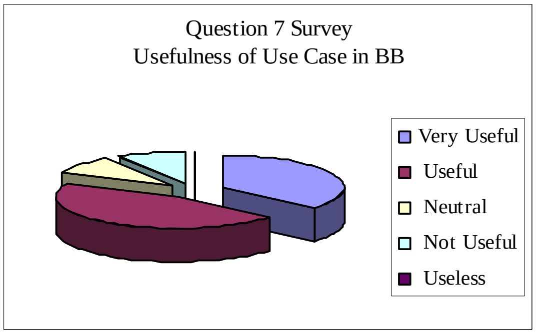


Figure 6.13 Relevance of FORM Business Model when developing BB

Comments by respondents indicated that the Business Models were mainly useful for (functional) domain analysis, and that when the BB(s) were finally modeled, they were somewhat independent of the business model. Thus the Business Model was considered useful in understanding the context of the functional (domain) but less relevant in the actual design of the BB themselves. This is consistent with the findings of question 5b (Figure 6.5).

6.3.5 Use Case Modelling for BB

When asked about the usefulness of the Use Case driven approach to developing BBs, respondents indicated very high satisfaction rating. Respondents identified use case diagrams and activity lists as particularly helpful, with supplementary requirements specifications quite useful. (question 7 & 8). These responses are presented in Figures 6.14, 6.15, 6.16 and 6.17



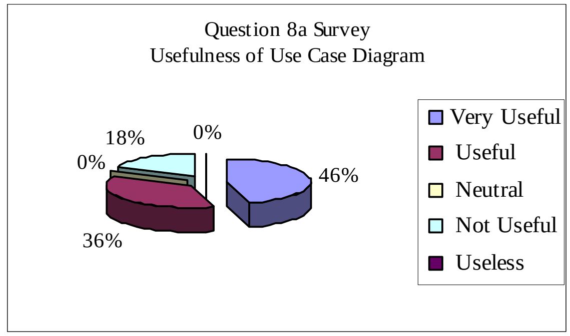
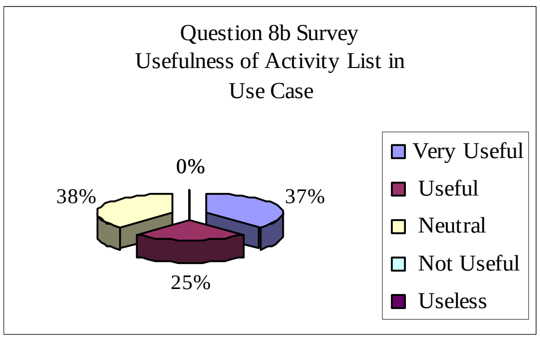


Figure 6.14 Usefulness of Use case Modelling

Figure 6.15 Usefulness of Use case Diagram



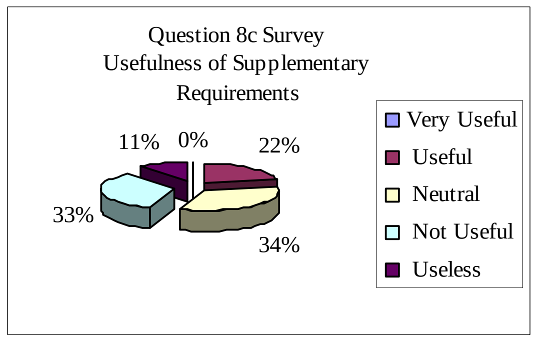


Figure 6.16 Usefulness of Activity List

Figure 6.17 Usefulness of Supplementary Requirements

6.3.6 Designing Building Blocks within or across the 3-tier architecture

Respondents were also asked to rate the use of the three tiers, namely: human, application (logic) and entity (storage), in the early analysis stages of identifying BBs (question 10). Respondents universally rated such an analysis level criteria from very useful to useful. No respondents objected to these criteria at this level. However, when asked about the same separation in the detailed development of a BB (i.e. to determine the boundary of a BB), respondents were evenly split. Some felt it counterproductive to attempt to split BBs across the three tiers. The reasons given generally involved the idea that a BB frequently required its own (internal) three layers – or at least two. This indicated the need for BBs to provide persistency as well as business logic. Another opinion indicated that BBs, to be reusable, needed to be less reliant on many database objects (persistency), but could manage the persistency themselves. Allied to this was the need to build business logic processing into these shared BBs, and thus they could arguably exist in either the business logic or persistency layers. The concern being that keeping the data and processing together would provide for a more self contained and useful Building Block. Splitting the Building Block into two BBs just because one can be focused on storage and retrieval whilst the other focuses on business logic could be very wasteful and cause unnecessary distributed interaction between the paired BBs. It was interesting that although developers originally thought that the BB would be separated into the 3-tiers, this actually was not the case in the final BB designs.

However, others argued the opposite (more traditional) case. These respondents argued that, for information objects which had to shared across BBs, applying the three-tier architecture would assist such sharing.

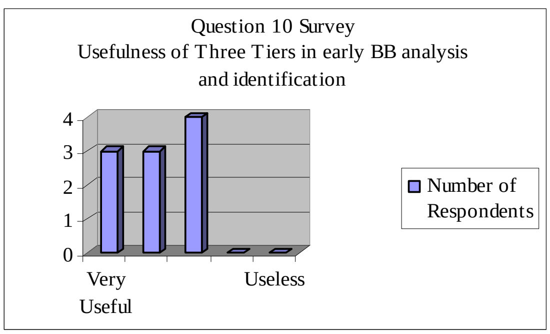


Figure 6.18 Three Tiers in Analysis

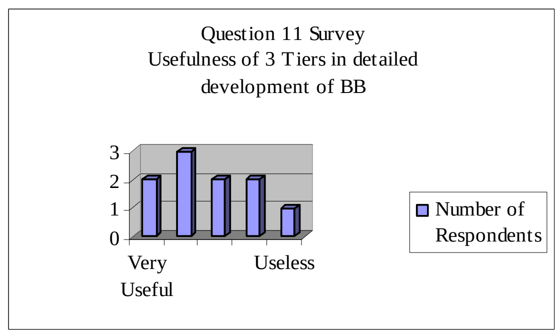


Figure 6.19 Three Tiers in Detailed Design

6.3.7 Key indicators to assist recognition of Building Blocks

A key aspect of the Building Block guideline was the identification of potential BBs during the second (and subsequent) iterations. Therefore, the evaluation asked the developers to indicate the usefulness of the following (questions 12a to 12f):

(a) Usefulness of the three tiers in determining the granularity or boundary of a BB

(b) Usefulness of the degree of self-contained behaviour in determining the granularity or boundary of a BB

(c) Usefulness of the level on inter-dependence between grouping of classes in determining the granularity or boundary of a BB

(d) Usefulness of ‘service’ that BB could uniquely support (in determining the granularity or boundary of a BB)

(e) Usefulness of potential unit of deployment in determining the granularity or boundary of a BB

(f) Usefulness of potential of BB as a unit of manageability in determining the granularity or boundary of a BB

The developers were asked to rate each criteria for identifying BBs and BB Contracts, on a scale from 1 (useless) to 5 (very useful). Table 6.1 gives their average opinion for each criteria and the standard deviation across the opinions expressed for each criteria.

Criteria for determining Building Block	**Average**	**Standard Deviation**
Usefulness of potential unit of deployment in determining the granularity or boundary of a BB	4.09	0.83
Usefulness of service that BB could uniquely support (in determining the granularity or boundary of a BB)	3.91	0.70
Usefulness of degree of self contained behaviour in determining the granularity or boundary of a BB	3.91	0.83
Usefulness of potential of BB as a unit of manageability in determining the granularity or boundary of a BB	3.55	1.44
Usefulness of three tiers in determining the granularity or boundary of a BB	3	1.48
Usefulness of level on inter-dependence between grouping of classes in determining the granularity or boundary of a BB	2.91	1.22

Table 6.1 Usefulness of Criteria for determining Building Blocks

The survey indicates quite reasonable agreement across the top three criteria, but wider disagreement over the latter three criteria. Although the population of the sample size is too small (11 BB developers answered questionnaires) for statistical significance, the results are interesting and act as a validation of the guidelines.

6.3.8 Modelling Building Blocks

With regards to modelling, the BB using Use Cases, Object Models, Collaboration Models and Interface Signatures, the developers reported this to be quite easy and useful.

Figure 6.20, 6.21, 6.22, 6.23 indicate the usefulness of the guidelines describing and modelling BBs.

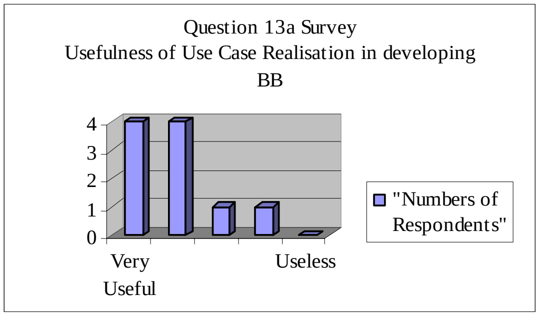


Figure 6.20 Use Case Realisation Modelling

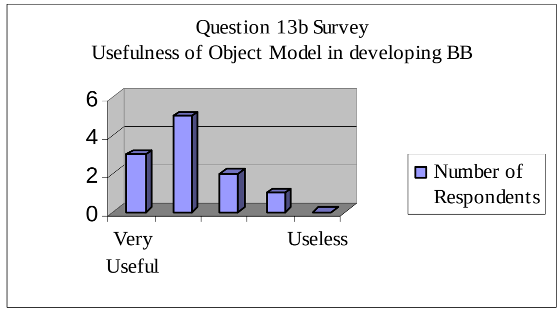


Figure 6.21 Object Model (for BB external information model)

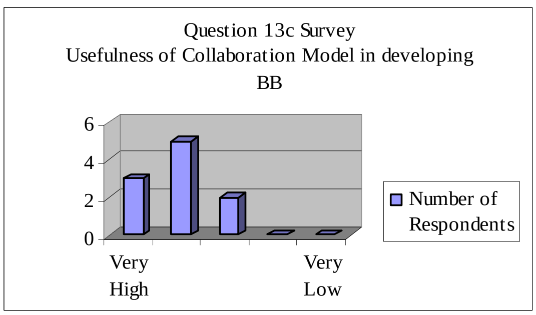


Figure 6.22 Collaboration Modelling

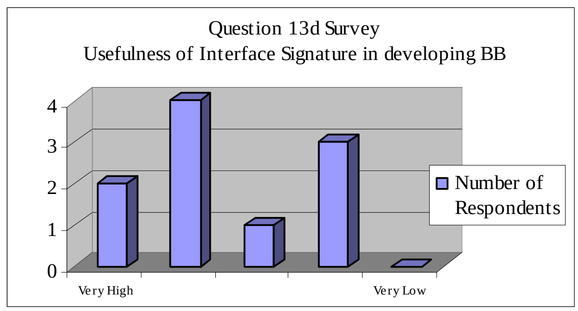
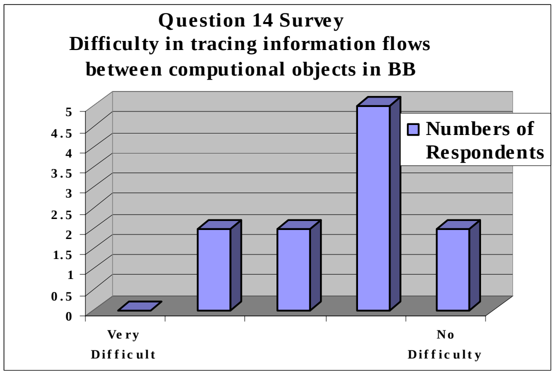


Figure 6.23 Interface Signature Modelling

An interesting point was the developers finding the interface signature not particularly useful for their modelling effort.

Difficulty in co-coordinating tracing information between computational objects within BBs and external to BBs

The developers were asked about difficulties in modelling (tracing) information within the BB as well as tracing information which was needed by their BBs but was located in the shared (external) information models. The responses show little difficulty, particularly with modelling internal BB information. The external BB information model was considered more difficult as this was defined against both available (standard) information models as well as BB information requirements.



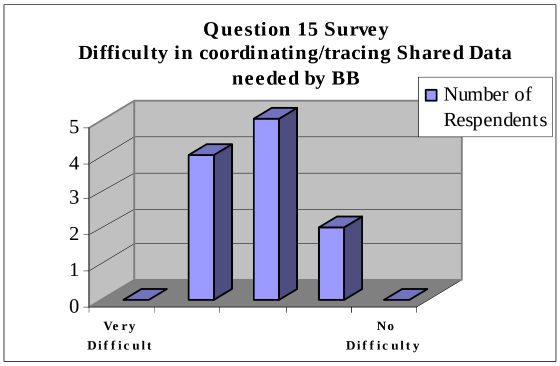
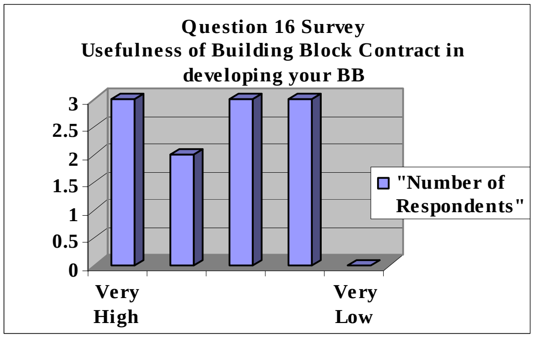


Figure 6.24 Collaboration Modelling

Figure 6.25 Interface Signature Modelling

Difficulty in developing BB Contract and BB description

Finally, the developers were asked about the usefulness of developing Building Block Contracts and describing their BBs. The interesting result about the usefulness of the contract is the range (spectrum) of opinion. It was almost evenly spread between very high and low (on average the developers rated it as useful/very useful. The description of the BBs themselves was considered quite easy.



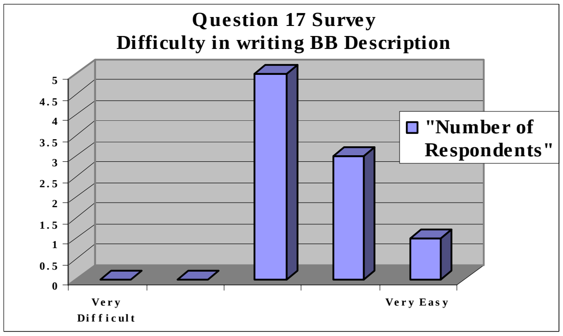


Figure 6.26 Usefulness of Modelling BBC

Figure 6.27 Difficulty in describing BB

6.3.9 Summary

Overall most developers indicate a high satisfaction level with the development approach for BB and found it useful in assisting the Development of re-usable software. In addition, most respondents found little difficulty in generating the specification of such BBs based on the guideline activities and artefacts.

Overall the key findings of the evaluation were:

(i) that the developers using the guideline were not reliant of prior expertise in RUP (and to a lesser extent UML) to successfully apply the guideline

(ii) the sample surveyed consisted of the appropriate developer roles and responsibilities intended for the guideline

(iii) the reference architecture (logical architecture) was very useful in providing context for the development teams. However the reference points and organizational boundaries were of less importance as the BB and BB Contract modeling work became more focused (BB Development phase)

(iv) the design activities and artifacts defined in the guidelines were very useful in describing and specifying the BBs and BB Contracts

(v) the Business modeling (activities & use cases) was more problematic of the developers. These were useful as a starting point for the BB and BB Contract use cases & activity diagrams but were not as relevant for the final BB and BB Contract specification

However, there are some caveats, which should be taken into consideration. Firstly, the actual sample size for the guideline evaluation is quite small (eleven developers), and the FORM project, although intended to develop pre-commercial prototypes, did not actually have real customers involved in the development. However, it should be also noted that the guideline was comprehensively used by the project partners and resulted in BBs which were subsequently used in several systems.

When asked for suggestions to improve the guidelines, the most common request was for more assistance in the construction and sharing of the information objects passed across BB boundaries (shared information model). This would correlate with the results of question 14 (Figure 6.24). Another difficulty identified was that the guidelines were being developed in tandem with the early stages of actual BB modelling work. This caused some difficulties early in the project until the guidelines because more stable however, this was addressed quickly within the first year of the project.

The final suggestion was in relation to the usefulness of reference points in the reference architecture. The suggestion was to de-emphasise the reference points for the later iterations of the BB development guideline as their full specification is less relevant to the BB modelling, but useful mainly for capturing the general business context of the application area.

6.4 Evaluation of Business Process Driven Development Guideline

The MODD Business Process driven guideline was evaluated as part of the second FORM trial that was conducted toward the end of the FORM project. The evaluation was primarily performed via questionnaire completed by the FORM development team. This team comprised of FORM partner managers, management system designers, management system developers (the majority of whom had not been involved in the BB design activities from the first trial). Some follow up interviews were then conducted where necessary to gain greater understanding of the questionnaire results.

The objective of the evaluation was to:

(i) Determine the prior experience/knowledge of the developers with respect to UML and RUP. And to determine their roles and responsibilities in the development process

(ii) Determine the developers own experience as to the usefulness and usability of the (BP) development guideline and the design artefacts produced by the guideline

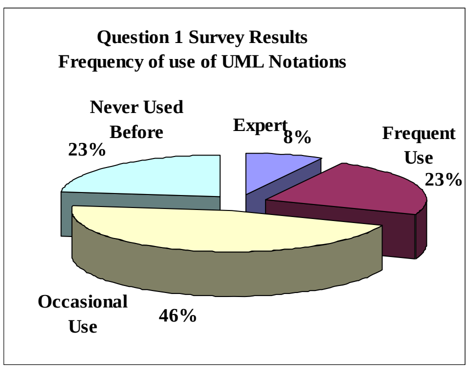
(iii) Examine the ease of modelling experience by the developers in applying the guideline

(iv) Identify any difficulties in using the development workflows or developing their design artefacts.

The questionnaire contained 25 questions, which examined the development teams experience in using the guideline. Thirteen completed questionnaires were analysed and the results of this analysis are presented below. Appendix 5 contains a copy of the evaluation forms for the BP development guideline and Appendix 6 provides a full set of graphs indicating the responses of the developers. The section below provides a discussion of these evaluations.

6.4.1 Prior Experience in UML and RUP

The survey asked two questions to ascertain the prior experience of the FORM development team's experience of using UML and RUP. Over 63% of participants indicated either no prior or only occasional use of UML, with approximately 30% indicating the never used before, and 30% indicating frequent use. Only one respondent claimed expert usage of UML. This shows that the development teams were not particularly skilled in UML modelling before the business process development.



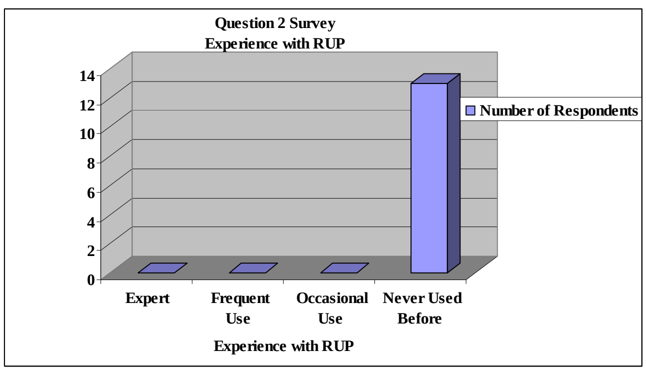


Figure 6.28 Experience in UML **Figure 6.29 Experience in RUP**

When asked about the prior usage of RUP, 100% respondents indicated that they no prior experience in this development process. Subsequent investigation indicted that several of the FORM organizations for whom the developers work, had employed RUP but this use was either project specific or only used by a very small set of business units within the organization. It is important to note that the BP Guideline was not based on RUP. However, as RUP is a widely used commercial OO development process, (and some of the Guideline workflows are loosely based on RUP), the lack of prior RUP experience indicates the learning curve which was required by the system developers.

The development team's wide lack of experience in RUP and UML was one of the issues, which the Guideline had to tackle, both in the presentation of the Guideline and in the exemplars & workshops used to instruct partners in the use of the guideline.

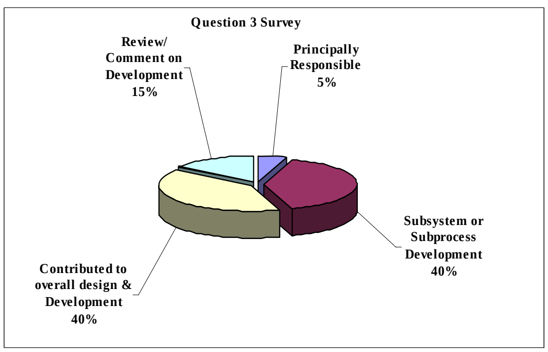
6.4.2 Responsibilities and Roles in BP System Development:

From the questionnaire results, the two principal roles taken by those developers surveyed were:

(i) Responsibility for sub-system or sub-process development,

(ii) Overall contribution to design & development.

The first of these indicates a sizable, responsibility for the full development cycle of a subsystem or sub process, the second indicates a more general contribution to design/development but not responsibility for that subsystem/sub process itself.



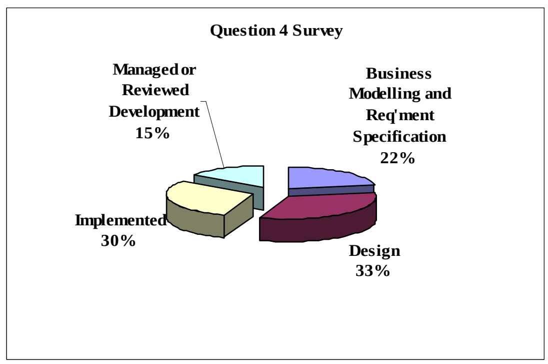


Figure 6.30 Responsibilities for BP System Development

Figure 6.31 Role in BP System Development

Question four broke down the principle development roles that the developers played in the BP development as: Business Modelling/Requirement analyst designers, Implementor, Development Managers/Reviewers. Once again the design and development roles were very heavily represented with 70% and 61% of respondents indicating they took these roles within the development process. 46% acted in the role of requirements analyst and 30% as development managers/reviewers. As would be expected, developers played multiple roles during the development process, with 53% of respondents playing 2 roles, and 23% playing only one role. Only 7 % (one respondent) claimed to play all four roles.

Subsequent investigation revealed that the development roles were principally that of system analyst, software developer and systems integrator. Few of the respondents in the development process were business analysts in their own organisation, however many had acted in senior system architect roles and had wide experience in the early design of commercial products and research prototypes.

6.4.3 Usefulness of Reference Architecture Workflows in BP Development

One of the key aspects of the BP Methodology was importance and usefulness of the artefacts (e.g. models) which the development process prescribes e.g business model, Reference Model. Question 5 asked about the usefulness of the Business model (Figure 6.32).

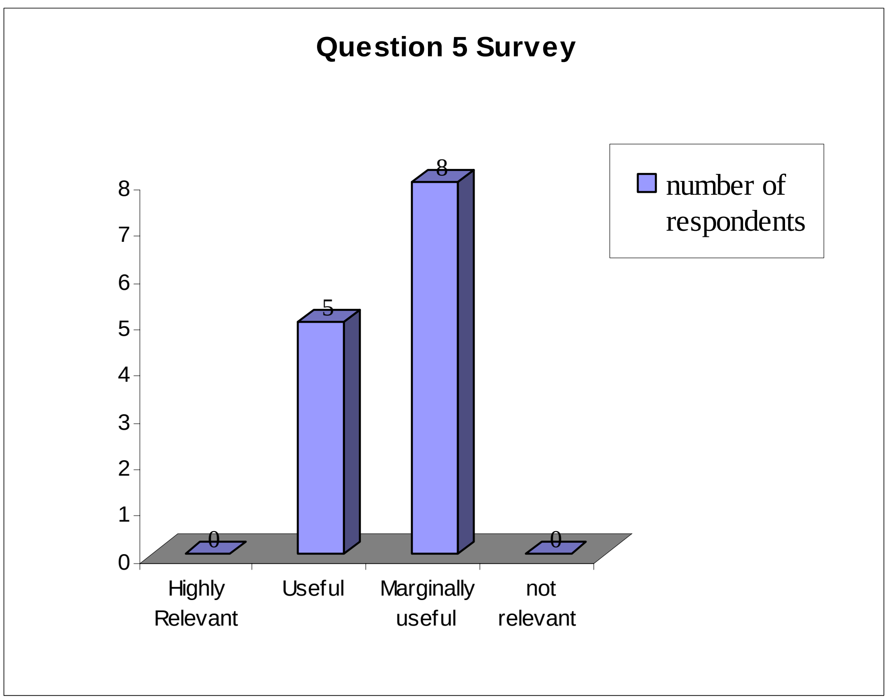


Figure 6.32 Relevancy of Business Model for developing BP system

Almost 40% of respondents found the specification of such a reference model useful, with 60% of respondents finding it marginally useful. However none found it highly relevant and none found it not relevant. A frequent comment regarding the reference architecture was that it was useful in setting the overall context of the various business processes under development, in indicating their ‘ownership’ (i.e. stakeholder within which the business process would execute) and in identifying potential cross stakeholder and inter business process communication. One such respondent who only rated it marginally useful indicated his reasoning as “it was only useful for (specifying) the interface definition” Respondents indicated that the reference model was most useful at this level of abstraction.

6.4.4 Usefulness of Reference Architecture, Business Modelling Workflow and Artefacts

Question 6 asked the respondents to rate the usefulness of the Business Modelling Workflow. In particular, it asked about the development activities defined in the guideline to specify the Reference Architecture, Business Use Cases and Business Activity Diagrams. Figure 6.33 indicates the relative experience for each of the modelling activities and artefacts.

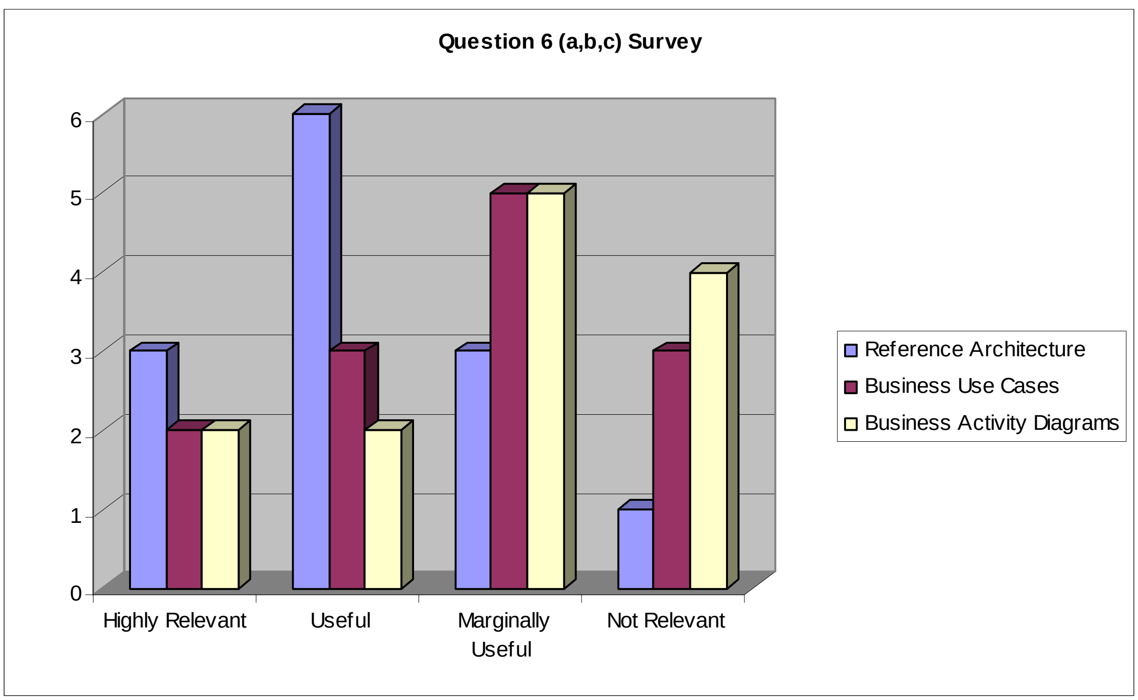


Figure 6.33 Relevancy of Reference Architecture, Business Use Cases and Business Activity Diagrams

The results for the Business Modelling Workflows relevancy were (in percentages):

Reference Architecture:	23% highly relevant, 46 % useful, 23% marginally useful, 7% not relevant
Business Use Cases:	15% highly relevant, 23 % useful, 38% marginally useful, 23% not relevant
Business Activity Diagrams:	17% highly relevant, 17% useful, 38% marginally useful, 30% not relevant

From the results, there is a clear drift in the usefulness of the business modelling workflow with 69% of respondents indicating the reference architecture workflow either highly relevant or useful, compared to 34% indicating the business activity modelling workflow highly relevant or useful. Many respondents indicated that the reference architecture was very important in order to understand how specific functional management application areas may interact and co-operate with other functional areas. One respondent commented, "the reference architecture proved to be especially useful in specification and description of the contract at the boundary between two domains".

Probably the most common difficulty cited by developers was the level of abstractness of the business activities. This would seem understandable as most of the respondents are not business analysts and therefore not particularly experienced in dealing with this level of abstractness. However, their opinion is important as such developers typically have to read business specifications and map them down to systems level designs and specifications. Also one of the developers who responded that the business models workflows and artefacts were 'not relevant' to ALL explained in his reason as 'I was not involved in this kind of work' – and thus indicated that they were irrelevant to him. Thus, at least in one of the BP systems, some of the development team were not involved in all aspects of the development effort. An interview subsequently revealed that this developer to be responsible for a subsystem which did not traverse an organizational boundary (i.e. was completely within one of the service providers).

It should also be noted that many developers commented on the usefulness of explicitly modelling the control flow between business activities. It seems that the difficulty was not so much with the business modelling workflow and the resultant activity diagrams, but rather with the level of abstractness of the business activities identified in the different development groups. Where finer grained business activities were identified, the business activity diagrams were considered more useful e.g. assurance, fulfilment/assurance. More specifically developers felt that in such cases where the business activities were more fine grained, they were very helpful in subsequently identifying appropriate BB Contracts.

Very few of the interviewed respondents had difficulty following the guideline workflow instructions and there was general positive agreement as to the form of the models generated by the workflow.

6.4.5 Requirements Engineering

These questions focused on the usefulness of the guideline's representation of requirements and the engineering of those requirements. In particular the questions asked about the usefulness of the system use cases in specifying information flow requirements of the BP system (question 7), the importance they attached to the use cases, activity diagrams & specifications and supplementary requirements documents (question 8a, 8b & 8c).

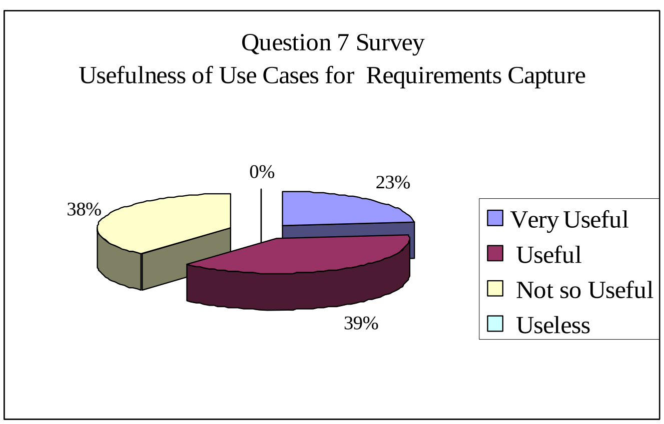


Figure 6.34 Usefulness of System Use Cases for capture information flow requirement

From the answers in Question 7 (Figure 6.34), 63% found the use cases useful or very useful.

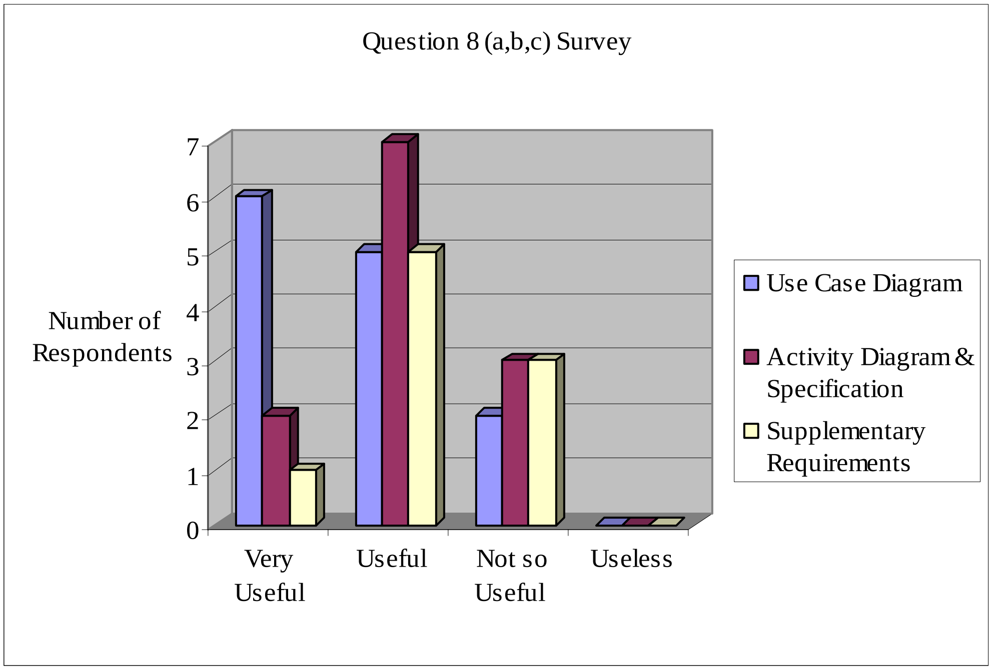


Figure 6.35 Usefulness of use case diagram, activity diagram & specification and supplementary requirements

From the answers, (in Figure 6.35) it is clear that use cases and activity diagram specifications were found very beneficial (i.e. approx. 85% of participants regarded the use case modelling activity as very useful or useful and 70% of participants regarded activity modelling as very useful or useful). Comments from participants indicated that these were easy to create, read and were useful in discussion of the requirements. The supplementary requirement specification activity was also found to be useful (66%).

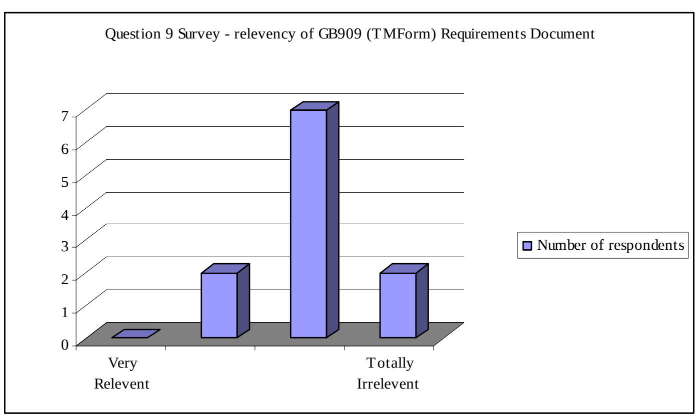


Figure 6.36 Relevency of GB909 (TeleManagement Forum) Requirements Document

When asked to indicate the relevance of the GB909 requirements[1] in performing the modelling work, 18% rated them irrelevant, and 63% rated them only slightly relevant. Typically, participants believed they were not that relevant to modelling work. In addition, as each partner was focused on a set of business processes, large parts of GB909 were irrelevant to them. Some other participants indicated that many of the requirements just stated general guidelines, which are so well accepted as to be superfluous and did not add any extra guidance. Those who rated the requirements as relevant highlighted the fact that such requirements reflected the non-functional requirements of the systems under development.

6.4.6 Separation into 3 tier architecture in early analysis stages of design

One of the issues discussed during the design of the BP driven systems, was the use of a three-tier architecture. When questioned as to the usefulness of this separation when developing the BP based systems (during the analysis workflows of the guideline), the developers indicated a some divergence of opinion. 69% rated such separation as very useful or useful in the analysis stage, where as 23% rated it as very poor (i.e. useless). It is perhaps a little surprising as such divergent opinion. Figure 6.37 presents this range of opinion.

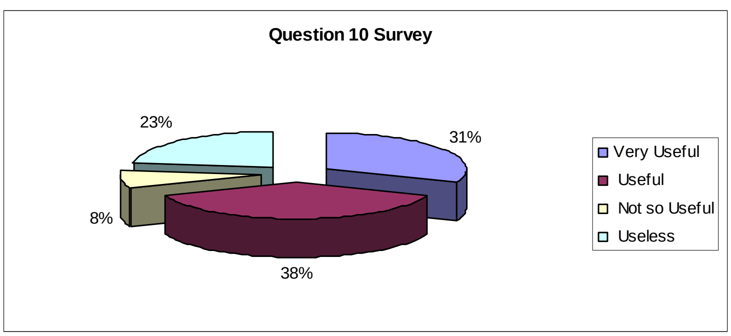


Figure 6.37 Usefulness of separation of activities into three tiers in the early analysis of BP based system

[1] GB909 requirements is a generic requirements specification for telecommunication service management systems produced by TeleManagement Forum

Some respondents felt that the benefit of the three-tier architecture "proved useful in the design of the workflow (process) …. It helped centralize the process logic, so modifications and updates to the system were easier to achieve."

6.4.7 Modelling activities for Control flows and Data Flows

Questions 11 and 12 focused on the modelling of systems processes – in particular, the representation of control flows and data flows (Figure 6.38). 90% of participants rated the system activity modelling work as very good or good for representing control flow logic in the system. No participants rated the activity poorly.

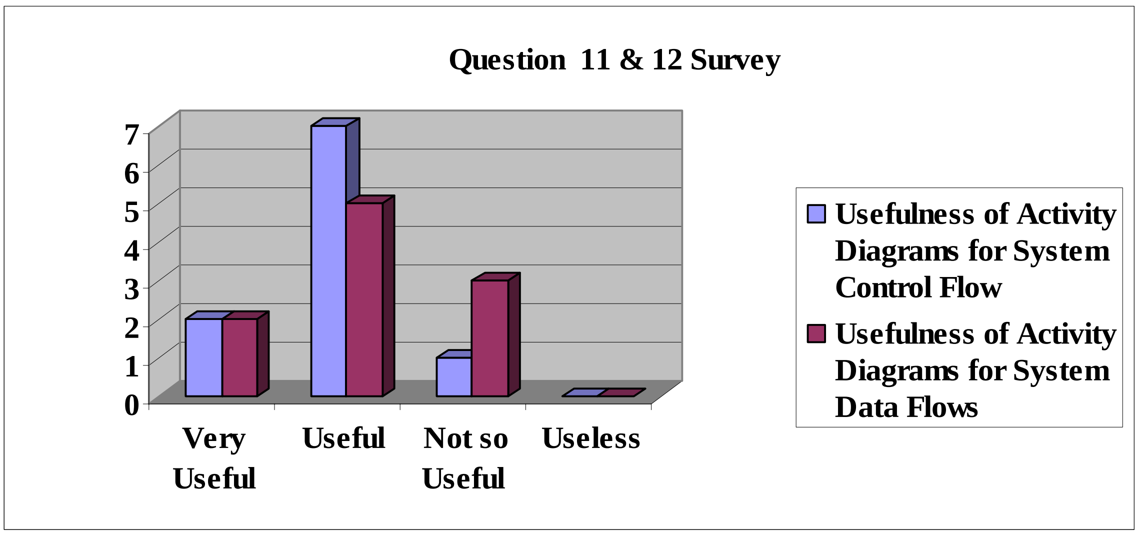


Figure 6.38 Usefulness of activity diagrams to capture the control and data flows for System

Comments by the participants indicated that the activity was very useful as a prelude to integration of the BBs (to support the BP driven system). Some participants preferred to use a collaboration diagrams rather than an activity diagram, to capture the control flow logic. This is possible where all control logic is modelled as activities (or collaborating object) and where no logic is external to such collaborating objects. Another comment stated that the activity diagram was good at capturing basic control flow but was not expressive enough to capture flows that are more complex e.g. multiple invocation of an activity. In such cases, the complex control flow was modelled as an activity.

70% of participants rated the data flow modelling activity as very good or good, again with no participants rating it poorly. Most participants found the data flow modelling straight forward and helpful. One participant did not use an activity diagram to represent data flow, but rather used a collaboration diagram, which indicated data objects flowing between the collaborating activity objects. One participant indicated that complex data flow could not be captured using activity models, and that an activity was used to represent the mapping of information from one activity to another.

6.4.8 External Information Modelling

Approximately 60% of participants rated the external information modelling as poor or very poor (Figure 6.39). Comments from the participants indicated that they believed in some cases the EIM was too abstract, others thought it complex and unusable. Others indicated that the EIM would only be useful to internally help define the contracts of the BBs. Others indicated that transformation activities would be required anyway, so that a consistent, universal EIM for the framework would probably be too difficult to achieve. Others felt that a federated approach, with no universal EIM, would be more appropriate.

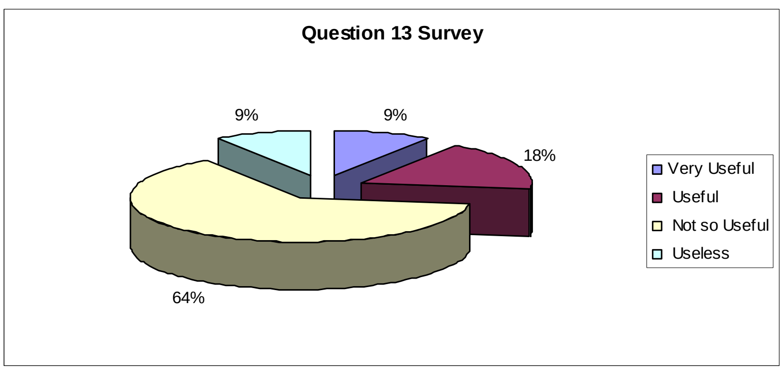


Figure 6.39 Usefulness of EIM in representing the shared information

It is clear, talking to the participants, that a significant influence on the EIM used by participants was based on the chosen application area, e.g. DMTF CIM for assurance, IEFT and ITU for fulfillment and IPDR for accounting. To attempt to integrate these information models was believed to be beyond the project and something, which may not be worthwhile. Agreement of some basic common information was useful e.g. SLA, etc. However, even for these information objects, transformations were sometimes required.

6.4.8.1 Mapping System Activities to Building Block Contracts

59% of participants rated this activity as being performed with little or very little difficulty. 33% of participants rated this mapping as difficult. Participants found that they

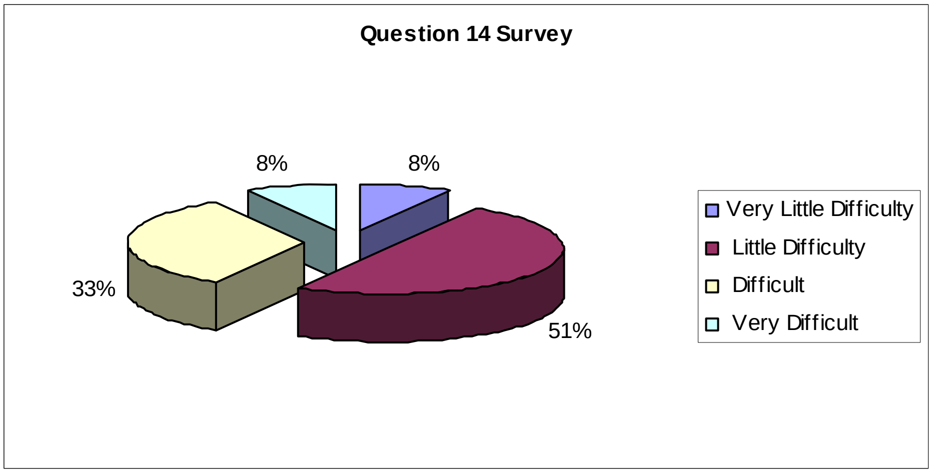


Figure 6.40 Degree of difficulty in mapping system activities to pre-defined BB Contracts

could easily handle any extra modelling required to accommodate the use of the BBs. Figure 6.40 depicts the degree of difficulty experienced by the development team in achieving this mapping.

Some participants indicated that as the BBs were developed within the same organisation as the business process system, the degree of difficulty is perhaps a little flattering. Another indicated that some re-design of the BB originally envisaged was required. This was needed to allow the business logic to be externally invoked rather then be performed completely within the BB.

6.4.8.2 Mapping System Information Flows onto BB Contract Boundaries

50% of participants rated this activity as being performed with little or very little difficulty, with 50% indicating it was difficult. Figure 6.41 presents the overall percentage responses from the BP developers. No participants rated it as being very difficult. The difficulties experienced tended to involve data transformation (between BB

Contracts) and the modelling of extra activities needed to either retrieve or store missing information. However, the participants found they could readily recognise when such design activity was required and were able to deal with it in a systematic way.

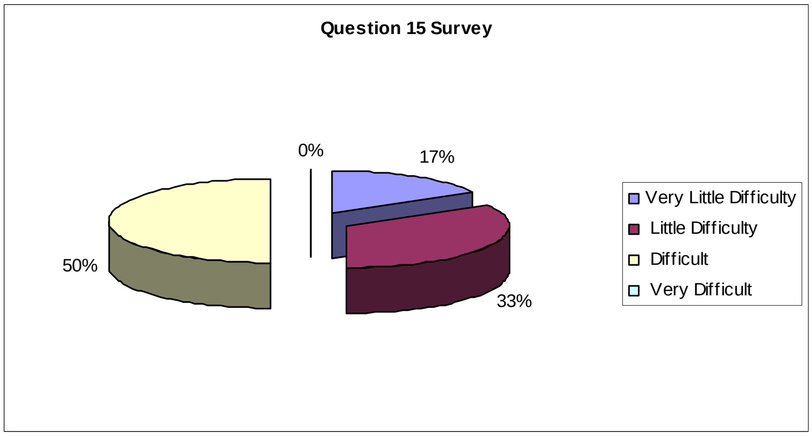


Figure 6.41 Degree of difficulty in mapping system information flows onto BB Contract boundary information models

6.4.8.3 Collaboration Modelling of BB Contract Interactions

Approximately 64% of participants indicate little or very little difficulty in performing this workflow (Figure 6.42). No participants rated it as very difficult. Comments from participants indicated that this diagram was very useful in communicating the BB Collaborations, which provide the basis for the Business Process Implementation.

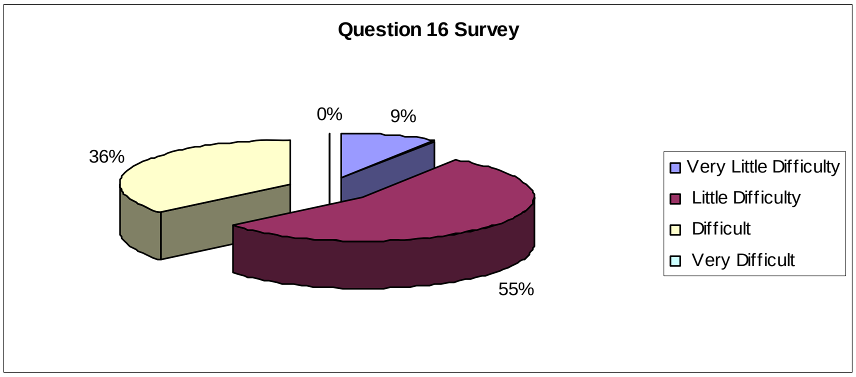


Figure 6.42 Degree of difficulty in designing collaboration diagrams representing BB Contract interactions (based on system activity diagrams)

62% of participants indicated that they were very satisfied that such collaboration diagrams provide enough information regarding the interactions between BB contracts (to support the system process to be implemented).

6.4.8.4 Mapping BB Contract collaborations on to BB Collaborations

This question investigated the difficulty in mapping BB Contract Collaboration diagrams onto BB Collaboration which could support them. 60% of respondents indicated that this modelling activity was performed with little or very little difficulty with 40% indicating some difficulty. No respondents indicated this task was very difficult (Figure 6.43).

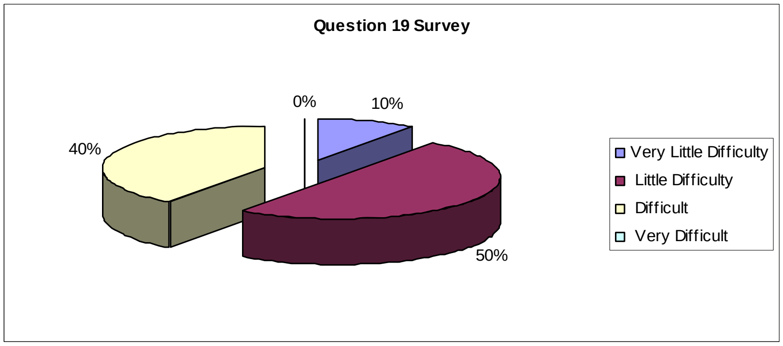


Figure 6.43 Difficulty in mapping BB Contract Collaborations onto BB Collaborations (which could implement them)

6.4.8.5 Opinion of Methodology/Guideline Description

Most participants (54%) indicated that only between 1 and 5 hours were needed to study the guideline). However, 30% of participants spent more that 5 hours studying the guideline (this was in addition to several guideline presentations and tutorials given during the project). Figure 6.44 indicates the relative time spent by developers in studying the guidelines.

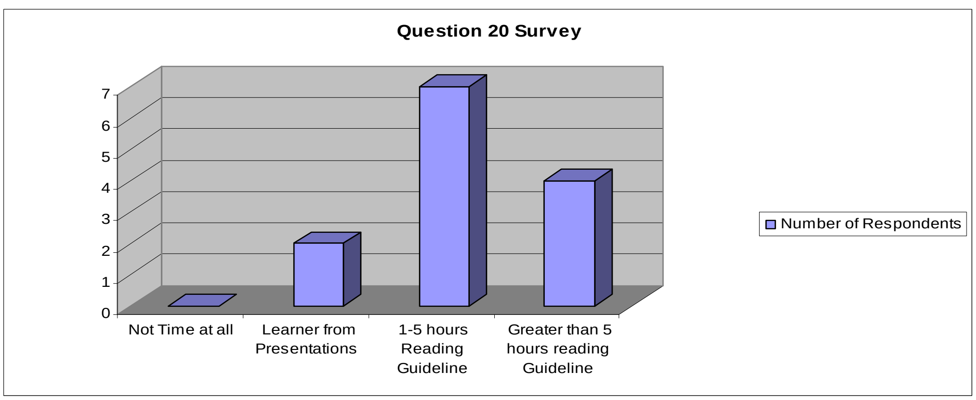


Figure 6.44 Time spend studying BP development guideline

All indicated that the combination of the workflow descriptions as well as the worked example(s) were very useful in gaining a better understanding of the guidelines prescribed activities.

Participants were asked to rate the clarity of the guideline in describing the various workflows in the development lifecycle (Figure 6.45). Overall approximately 60% of participants rated the guidelines as very clear or clear, while none indicated that it was very unclear. The only workflow which attracted any difficulty in understanding was the BB Contract Collaboration modelling activity. During the guideline development, this workflow was updated to give greater clarity to readers.

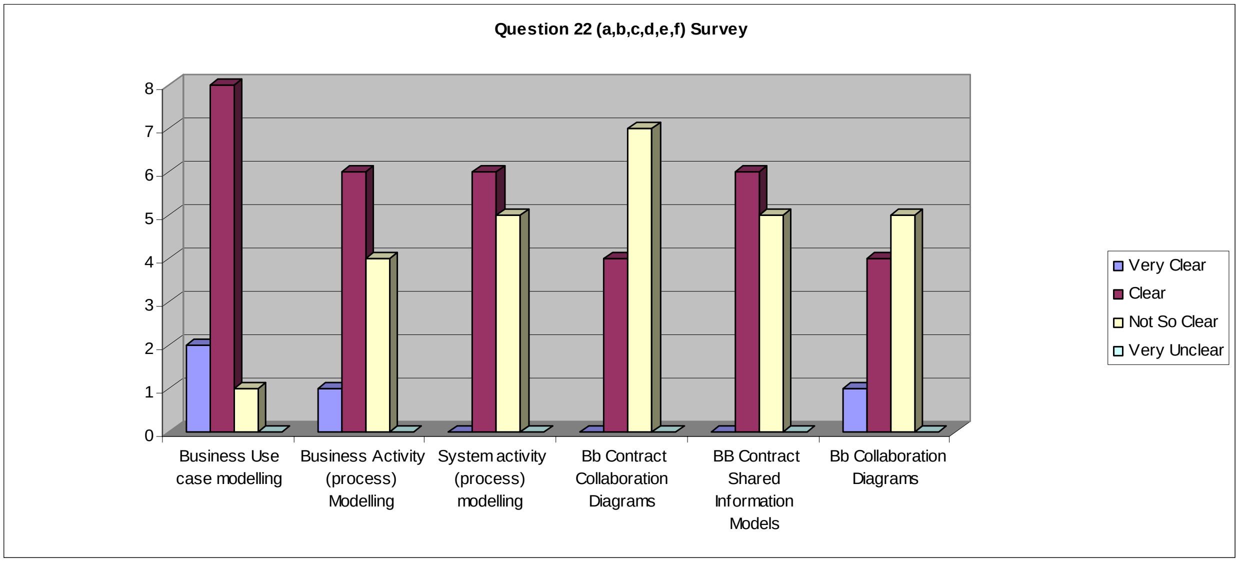


Figure 6.45 Developers ratings of development workflows for BP system

6.4.9 Summary

It is clear from the evaluation that the guideline has been successful in assisting all of the development teams in constructing their management systems. The principle advantages of the guideline seems to be its well formed set of development activities which provide tractability of artefacts and designs between the different workflows, and provides well explained development activities.

A summary of the key finding of the survey would be:

(i) No previous RUP experience and average UML experience. Thus the developers in the trial did not have any advantage in using the guideline based on prior knowledge.

(ii) The developers in the evaluation matched the intended audience of the guideline in most respects except for having expert business analysts.

(iii) Reference architecture was predominately seen as very useful in business analysis activities

(iii) Some developers had difficulty defining the business activities (based on the initial business use cases)

(iv) The system use cases and activity diagrams were seen as key for analysis and modeling of the systems. Also these were rated very highly for capturing control and data flow (although collaboration diagrams were sometime used instead of activity diagrams for complex data flow).

(v) Probably the main difficulty experienced by the developers was the modeling of the Enterprise Information Model (a shared/federated model of the information required across the Business Process driven system).

(vi) The mapping of system activities to the pre defined BB contracts was very successful and did not create much difficulty

(vii) The mapping of system information flow to BB Contract boundary information models was more challenging. This was because some information items required transformations to allow the information flow to succeed.

Based on the developers survey, the areas where the guideline could be strengthened is in its treatment of complex control and data flows in activity diagrams. Developers alleviated the problem by simplifying the flows and by designing activities to handle the complexity or just preferred to use collaboration diagrams. Some issues remain about the content of the EIM, in particular, regarding the integration of heterogeneous information models from different standardization bodies. This is a general problem as there is frequently a need to map information from one standard model to another as the system traverses different management (functional) domains.

Like the BB evaluation some caveats of the evaluation need to be mentioned. Firstly the sample sizes for both evaluations were quite small (13) and therefore not statistically significant. Secondly, the developers were working on a research and development project and were not purely working on a product line. However, a subset of the prototypes was incorporated into ongoing product development work within some of the organisations.

6.5 Overall Summary of Key Evaluation Results

Overall the evaluation trials have indicated that both the Guidelines were easily followed and supported the development of reusable component design as well as construction of business process driven systems. The evaluations have supported the choice of dividing the development effort into two separate guidelines to reduce development process complexity. The evaluations have also indicated that an artefact based, agile development process in UML can provide an effective approach to component and systems development. The evaluations indicated that the MODD guidelines provided effective, step by step, advice for each development activity.

One aspect where MODD could be improved is support in aggregating or more easily integrating heterogeneous information models across the Building Block Contracts when support business processes. Although MODD supports for the definition of Building Block Contract boundary information models, it needs to provide techniques and guidance as to how such boundary information models can be aggregated (where necessary) to support seamless heterogeneous information flow across the required business (management) process. The evaluation highlighted this as being one area which the developers of the business process driven systems require more assistance. In situations where such boundary information models are drawn from the one information model standard e.g. IETF, DMTF's CIM, or TM Forum's SID, such aggregation is not too difficult. However, where a desired business process involves the integration of Building Block Contracts, whose boundary information models are drawn from multiple standards, or are defined in isolation to any agreed standard (or enterprise convention) such information model mapping can be difficult to achieve. One possible approach would be to attempt to define an ontology for the aggregated information models. Such an ontology could then be used to support mappings between semantically similar information objects and thus provide more predicable semantic information flow.

6.6 Related Work

This section compares and distinguishes the MODD research and guidelines from related work in the area. In particular this section compares and contrasts MODD BB development guidelines from RUP (upon which the BB Guideline was originally based). Then a comparison of MODD and TeleManagement Forum Lifecycle Methodology is presented. Finally the MODD guidelines are placed in context with the work of OMG Model Driven Architecture.

6.6.1 Comparing the Building Block Development Guideline with RUP

As RUP is intended as a ‘framework’ in which specific processes can be developed, this section examines how the BB Guideline could be considered as such a customization of RUP. The section also identifies how the BB guideline extended RUP by adding certain useful design artefact definitions and design activity workflows.

Implementing a system based on RUP can be achieved by authoring and adhering to a development case, which defines the specialisations/additions to the original RUP phases, activities, process workflows and artefacts. It is possible to present the BB Development Guideline as such a development case, which facilitates the development and specification of BBs and Building Block Contracts. In particular such a development case would focus on tailoring the various aspects of RUP’s business modelling, requirements modelling, analysis and design, implementation and deployment workflows.

The Context Modelling Phase of the BB Development Guideline can be thought of as a customization of RUP that selects and customises the activities that are to be performed and artefacts that are to be produced as well as adding additional elements where necessary. However, for the guideline, it was found that RUP was lacking in support for a “component design”. In RUP, a component is a runtime unit of software, whereas in the desire is to provide a reusable grouping of models and artefacts (a design time unit of reuse), which is rich enough to adequately represent the properties, pre- and post-conditions, constraints, potential usage, and interfaces which is termed a Building Block Contract. A Building Block may support one or more Building Block Contracts. A FORM Building Block more closely resembles a “Sub System” in RUP.

Thus the BB Guideline customises the RUP phases to reflect the goal of developing BBs and Building Block Contracts for reusable service definitions and design time components which can support them.

6.6.1.1 Mapping the Guideline Development workflows into an extended RUP

The BBs Development Guideline can be presented as an extension and customization of several aspects of RUP. RUP was designed for a broad range of application and enterprise systems. However, this guideline is not focused on application development, but rather reusable component development. Therefore augmentation and extension of various parts of RUP were necessary because of the development target. For example, because the guideline is focused on the development of management components, effort is needed to initially model the problem domain in order to identify and scope candidate components as well as to design and specify the components (once they are agreed). Also, management systems (for service and network management) are frequently distributed across organisational or administrative boundaries. Such management system boundaries are usually represented in the telecoms area as reference architecture[1]. Interactions across such boundaries in the reference architecture become sets of coarse-grained integration points for (possibly multi-vendor) software components. .

Additionally, other differences between the BB development guideline and RUP involve changes to the RUP Development Phases, Model & Artefact Selection, Process Workflows and Documentation.

Development Phases & Iterations.

The Context Modelling Phase for the BB Guideline is more extensive than would normally be the case for general application of the Inception Phase of RUP. The Business Modelling work is important in initially scoping the management process areas and domain responsibilities for which the BBs will provide support. The Business Modelling work involves identifying the domain boundaries of the envisaged management systems

[1] Different standards bodies have used the notion of reference points to assist in describing relationships between either business entities or systems e.g. TINA.

and the organisation in which they may reside, identifying management use cases and initially modelling the management processes which support them. An additional outcome of this phase is the modelling of analysis objects, which support these use cases and management processes. These analysis objects are used in the BB Development Phase in the identification and determination of candidate BBs. Also this phase allows the modelling of reference points between Business Organisations identified during business modelling.

External influences in this phase include the adherence/alignment with relevant telecom industry standards e.g. TeleManagement Forum's Telecom Operation Map (and F-A-B processes) [TeleManagement Forum eTOM 2004]. Such standards can have a significant influence on the management processes being modelled as well as on the information (object) model(s) associated with these processes. However, it is important to remember that the objective of this development Guideline is the development of BBs and Building Block Contracts. The effort therefore in this (context) phase is to identify the business processes, actors, potential organisational boundaries, etc. which impact on and place in context the BBs and building block contracts which will be subsequently developed in the later development phases.

The BB Development Phase focuses on the development of specific BBs and Building Block Contracts and involves testing of these BBs. This involves the identification of BB and the re-organisation and grouping of analysis objects into re-usable BB designs and specifications. This phase also involves the modelling of information used both within the building block(s) as well as those (information objects) passed into or communicated out of the BB. The information communicated by a Building Block Contract is termed its 'Boundary Information Model' and is documented as an explicit information model indicating the informational requirements and outputs of the Building Block. The key output of this phase is the specification of Building Block Contracts and the design of BBs, which support them. RUP does not provide support or provide atrefact definitions for modeling (BB) Contracts and (BB) Boundary Information Models.

Another subtle but key difference between RUP and the BB development guideline is that the BB development guideline (in the BB development phase) just focuses on the modeling of one or more BB and BB Contract. This allows the BB Guideline to be more 'component' driven and provide a more 'incremental' development of the BBs (and BB Contracts), rather than developing the 'system' as a whole (which is the RUP approach). Thus the BB Guideline's concentration on being a lightweight, incremental development process (for BBs and BB Contracts) would align the guideline more as an 'agile' development process rather than traditional object oriented development processes.

The RUP Construction Phase and Transition Phase are not considered within the BB Development Guideline.

Models and Artefacts Selection.

RUP defines a very large set of potential modelling and documentation artefacts. The BB Development Guideline prescribes a reduced set of models necessary to design and specify BB. In addition, the guideline specifies a Building Block Specification Template, which presents a rich description of the Building Block, essential for later reuse.

Two additional artefacts, not in RUP, are contained in the Guideline, namely:

(i) An explicit architectural (structural) diagram, called the Reference Architecture, which indicates boundaries between management/administrative boundaries and the placement of business processes within and/or across these boundaries. This architectural diagram attempts to reconcile the management process areas (for which BBs are to be developed) with the organisation or administrative boundaries being considered within the Business context.

(ii) The explicit modelling of BBs and specification in XML of the Building Block Contracts. These Building Block models are considered fundamental to the FORM Framework and do not have a direct equivalent in RUP. The most similar package of models defined in RUP, which includes the grouping of such required models, is that of a Sub System package.

Process Workflows

The Guideline concentrates on the Process Workflows (rather than the support workflows). This does not mean such support workflows are not relevant; on the contrary, they are very useful. However, as this guideline is focused on the actual development activities (rather than their management or product deployment), the Process Workflows are the only activities customised. This customisation is necessary for the development of BBs and the modelling artefacts that are used to specify them.

Documentation

The guideline focuses on only a few key document templates, including a vision document, software architecture and building block specification.

6.6.2 Comparing MODD Guidelines with TeleManagement Forum Lifecycle SANNR Methodology

This section provides a brief comparison of MODD with the TeleManagement Forum's lifecycle methodology. In particular it identifies some similarities and some considerable differences and discusses these under the headings (i) intended users (ii) objectives (iii) scope and (iv) methodology specification and depth. It is worth pointing out that the SANNR methodology was developed after the MODD methodology was published.

Intended Users

Both methodologies seek to support Service Providers and Service Developers.

Objective/Motivation

SANRR depicts a homogeneous based solution based on NGOSS and its primary aim is the use of eTOM, SID and NGOSS architecture for design and development. MODD's depicts a fundamentally heterogeneous service provider and service developer environment. It deliberately does not insist of a single standard or 'single consortium approach'. MODD is designed to be used with multiple Information Models (DMFT, IETF as well as TeleManagement Forum). MODD is neutral of the 'politics' of standards and tries to allow the service provider to make their own choices based on business or corporate needs. SNARR is deeply embedded in NGOSS, SIG and eTOM.

Scope

MODD and SNARR may initially seem to have similar scope (business modelling, systems modelling, etc.). However, a fundamental difference is that MODD provides development guidance for both business and system process systems development, as well as component and building block development. SANRR's guidance is in terms of identification that the contracts and business and system processes need to be defined and provides an indication as to where in the NGOSS specification the structure of a contract is defined. Thus it identifies that they should be developed and can only suggest an abstract set of approaches (Scope-Analyse-Normalise-Rationalise-Rectify) to achieve each. However, in SANRR there is no guidance as to how to achieve this. Similarly there is no guidance as to how these approaches will help 'document' these development activities. Finally, the level of depth of SANRR is currently very shallow and cannot seriously be intended as a system development process. It is useful in identifying what may be the basis of a methodology but its development is not mature enough to direct a development team in Contract or Business Process Systems design. It needs to be significantly developed to provide a more specific description of the starting points, development activities and outputs of each of the phases (steps) and views in the lifecycle [TeleManagementForum Lifecycle 2006].

Methodology Specification

The SNARR is depicted in a few simple diagrams (as illustrated in Chapter 3.3 and Table 3.1) and brief textual outlines of the activities. Although reference is made to existing methodologies in the background section of the SANRR description, the methodology provides no linkage to existing best practice in systems or information modelling and is wholly inadequate to support the development of a telecommunication management service without significant application of other development methodologies. This contrast to MODD, which takes a detailed, UML based, description of each of the design activities with clear starting points and outputs defined. MODD's approach which enhances existing development processes means that it is more easily supported by both international best practice and existing software development tools.

Conclusion

In essence, MODD is a much more comprehensive and usable methodology (than the lifecycle model in its present state). MODD is more general as it is not solely based in the usage of TeleManagement artefacts and models. Also unlike SANRR, MODDS usage has been tested and evaluated by teams of developers of management systems.

6.6.3 Comparison of MODD with PROSPECT Design Process

The PROSPECT Design Process defined a development cycle for components (computational objects) and management systems. The PROSPECT design process was one of the original influences for the MODD development guidelines. Thus it is not surprising that PROSPECT Design Process has some similarities to the MODD development guidelines in that the PROSPECT process supports the basic notions of business modelling (stakeholders), use case analysis (for requirements specification), definition of computational objects (groupings of classes).

However, the PROSPECT process mixes the development of the components with their integration to form the deployed management systems within a single development process. This is significantly different from MODD which separates the development of reusable management building blocks from the modelling and development of business process driven systems. The MODD separation of the business process driven system from the development of building blocks, allows much greater support for building block reuse. MODD also provides clearer, more explicit modelling of the business (management) processes. This separation reduces the complexity of each of the guidelines and focuses the guidelines on developing specific design artefacts (i.e. building blocks, building block contracts and management processes).

A second key different between PROSPECT design process and MODD, is the use of contracts and building block specifications to provide more flexible design (and reuse) of component design. The PROSPECT design process merely captures interface specification of component(s). The use of Building Block Contracts (and the models they encapsulate) provide greater support for both Building Block (contract) comprehension and reuse.

A third difference is PROSPECT's design process's failure to capture boundary or external information models needed for each component. In MODD these are explicitly specified to capture the Building Block and Building Block Contract's information requirements. These are not dealt with in the PROSPECT design process, which relies on an information model being shared across the PROSPECT components.

A final difference between MODD and the PROSPECT Design Process, is the use of UML to capture, explicitly, management system (business) processes. In MODD the Business Process driven development guideline provides guidance as to the modelling of such system processes and their mapping onto Building Block contracts (and hence their implementation via integrated Building Blocks). The PROSPECT Design Process focuses more on the deployment of the computation components but not on their design time integration.

6.6.4 Relationship with OMG's MDA

OMG's MDA proposes the idea of incrementally transforming Platform Independent Models into Platform Specific Models, from which executable code can be generated. The goal being that the programmer no longer needs to write programme code but just needs to develop models.

The MODD approach does not address the development or transformation to Platform Specific Models. However, the MODD could be used to guide developers in the design of the Platform Independent Models. Currently the models prescribed in the BB guideline are platform independent. Thus it would be an area of future investigation to examine the usage of the two guidelines to develop different kinds of Platform Independent Models, from which MDA transformation approach could then be carried out.

Some aspects of the business process driven guidelines may in fact be amenable for MDA type transformations. In particular the identification and mapping of system process activities and information models to BB contracts and external information models. It is unlikely that such mappings and integration could be performed completely automatically, but they may suit transformations where the developer can manipulate the system process models (information and activity). This is an area of possible future research and is outlined in the next chapter.

6.6.5 Summation of related work

This section has compared, contrasted and placed in context, the MODD guidelines with key methodological standards and initiatives in both general systems design as well as telecommunication service management design. The section has highlighted MODD unique contribution against these development approaches. The section has also identified possible synergies which could be investigated further in the future.

7 Conclusion

This chapter presents the conclusions of the thesis. First the chapter identifies the original objectives of the thesis and in particular discusses the achievements in defining the Building Block (BB) Development Guideline and the Business Process Development Guideline. The chapter then identifies the research contribution of the thesis. Finally the chapter concludes by identifying areas where future work would progress the research in this area.

7.1 Review of objectives

The key problems addressed by MODD was the need to reduce the complexity of the development effort, both in the development of service/components as well as the integration of these components into management applications; the need to support the possible automation in the development process (and hence increase the speed with which management services and applications can be developed); and the need to make best use of mainstream, technology neutral, development techniques (e.g. UML, MDA) whilst supporting the appropriate application of emergent standards e.g. TMF, DMTF.

The thesis proposed development activities required for both reusable component designs and for integrating their usage in business process driven systems development. To achieve these goals the thesis

(i) researched development approaches for telecommunciations service management

(ii) proposed and developed an integrated design methodology which supported (a) the modelling of (reusable) telecommunciation service management component designs, called Building Blocks (BB) and Building Block Contracts, and (b) a business process driven approach for telecommunication management systems development which supports the reuse of management Building Block and Building Block Contract designs.

(iii) validated the methodology with the development of (a) framework of telecommunications management BBs, BB Contracts and (b) business process based telecommunication system across a range of management application areas (namely fulfilment, assurance and billing).

The thesis first identifed the trends and developments in the state of the art in telecommunication management architectures and telecommunications management development methodologies. A comprehensive review of the key contextual drivers, architectural initiatives and methodological influences which are impacting the development of telecommunication service management today was presented. In particular, the thesis identified key drivers influencing development methodology. The thesis singled out the need to provide reduced complexity in the development of components and systems and the need to design system solutions which can be developed using existing component designs and services.

The thesis also reviewed the legacy and emergent architectures for future telecommunicatons management systems and identified the movement toward contract and service oriented development. From a methodological perspective, the thesis investigated current and state of the art methods for designing telecommunication management systems and provided a review of such methods.

7.1.1 Review of Building Block Development Guideline

The objectives of the BB development guideline were to:

- Guide the design activities in developing BBs Contracts & BBs.
- Specify the development workflows required to design the Building Block Contract.
- Identify modelling notations and the models to be developed during each development workflow. Indicate the traceability of artefacts developed across the development workflows.

- Prescribe sets of artefacts to characterize and communicate usage of Building Block Contracts.

The proposed BB development guideline defined the specific workflows, activities and artefacts needed to design BBs and BB Contracts. In particular, the guideline provided a customization and extension of RUP to facilitate the modelling and specification of BBs and BB Contacts using UML based artefacts specifically packaged for each. These workflows and design artefacts were presented and examples given of their development.

The guideline was evaluated via a user trial and was compared with emergent methodologies for telecommunication management systems. The user evaluation validated the guideline and provided feedback as to the usefulness of the artefacts developed, the design activities and the use of the workflows to iterate the development process. Overall, the evaluation was successful in both the development of BB Contracts and BBs. The guideline provided highly usable, lightweight development process which successfully supported the modeling of component designs and contracts. The guideline provided a easy to use development process, tailored to the needs to of the telecommunication component developers and reduced the complexity of using such industry wide development process frameworks such as RUP.

Some areas for improving the guideline were identified. Notably these involved the need to reduce the importance of modeling of reference points in the business model when design BBs and the need to provide greater support and clarity in the modelling of BB Contract External Information Models.

7.1.2 Business Process driven System Development Guideline

The thesis also proposed the Business Process driven system development guideline. The objectives of this guideline were to:

1. Support the modelling of Business and system processes
2. Support the refinement and mapping of business and system activities onto Building Block Contract definition(s) and associated information flows

3. Support the mapping of integrated (Business process driven) system designs onto existing BBs and mapping of information flows between BBs that support the business systems design. This also involves identifying occurrences where missing functionality is not supported by the BBs and support the development of object & information models where this functionality is missing.

In achieving these objectives the Business Process Development Guideline provided an iterative development cycle which supported eight stages of development. This development cycle included an easily followed process for: business modelling; business process modelling (for requirements capture and use case based design); system process modelling and system information modelling; system process re-modelling (to allow mapping of process onto BB contracts & mapping of system processes onto BB contracts); mapping of system process information model(s) to BB Contract External Information Models; modelling of information and behaviours not supported by BB contracts; mapping of BB contracts to actual BB designs; systems integration and performance of testing and deployment. Because the business process development cycle has a focus on modelling rather than implementation, the last two stages of the development cycle were less well specified.

The evaluation of the guideline illustrated the success of the guideline in modelling the systems and assisting in the reuse of the BB Contracts and BBs to realise these systems. The evaluation showed the usefulness, clarity and applicability of the workflows, activities and artefacts defined in the guideline. Overall the guideline was very favourably evaluated by the developers in the trial. A few areas of guideline improvement were identified, principally in the area of modelling complex information flows and control flows in activity diagrams.

7.2 Contribution of thesis

The thesis' principle contribution is its reduction of the complexity of developing both reusable components as well as developing systems from these components. By separating the methodology into two inter-related guidelines, each guideline was able to focus on the core needs of its intended users, further decreasing the complexity of the development effort. The guidelines were able to blend the use of best practice in software modeling with the introduction of new development phases, activities and artifacts. This provided a novel enhancement of existing development practices without the need to develop newer modeling tools and environments.

The guidelines succeeded in defining lightweight, easy to follow workflows, design activities and design artefacts that provided a clear development methodology whilst reducing development process complexity. The guidelines promoted the reuse of design time BBs and building block contracts, to encourage relatively easy utilization in business process driven system development. The two guidelines also support the different communities emerging in the service operation and component vendor markets.

This combination of the two guidelines (and their development cycles) provide an innovative way of conducting systems development and design reuse. An ancillary contribution of the guidelines is that they facilitate component and system design across a range of management standards and information models. The guidelines are also particularly suited to telecommunication service management because of its twin influences of the need for reusable management services and the telecommunications industry's keenness on business process driven approaches.

The generality of the MODD methodology is demonstrated as it can be applied to many different domains within telecommunication service and network management. The trial systems, which were used in the evaluation, ranged from customer management, service provision, accounting management, security management, performance management, web application management, as well as network management and IP quality of service management.

Thus the key benefits of MODD are its reduction in the complexity of the development process (as compared to such development processes as RUP), and its focus on iterative development with clearly defined development workflows & explicit design 'artefacts' at each stage of development. This reduction in complexity should enable more rapid development of reusable components (or services) and business process driven systems.

It should also result in less unnecessary documentation and modelling effort, as the guidelines specify a reduced set of workflows and UML modeling artefacts.

Another key benefit for the development stakeholders (i.e. component developers and system developers) is its separation of the development effort in to two explicit (but integrated) guidelines. The separation of key concerns of component (or service) developers and system developers provides more focused, easy to follow, development activities tailored to the needs and pressures of these stakeholders.

7.3 Future Work

The research in this thesis can be progressed further in several ways. Firstly the work could be enhanced by applying UML profiles to the BB development guideline. This research would seek to support domain specific profiles which would be more amenable to automated programming code generation systems which are beginning to emerge in the Model Driven Development arena. Such profiles could be based on the particular platform technologies most prevalent in telecommunications operators management centres. This research could also enhance the guidelines to support UML 2.0.

A second area for research would be the inclusion of policy modelling and development as part of the BB and BB Contract design. Policies are event-condition-action type constructs which allow a component or system to offer adaptivity. The research would focus on the design activities and workflows needed to model policy interfaces for a BB so that it could be sent policies at runtime to affect their behaviour. The policy would thus support a management interface for BB and be accessible via a BB management contract.

A third area of future work would be the profiling of MODD to support specific standardisation efforts. For example the MODD guidelines could be very useful in the development of purely TeleManagement Forum compliant systems and components. Although MODD is neutral to either TeleManagement Forum or DMTF, MODD could be profiled to focus exclusively for the development of TeleManagement Forum technology neutral Contracts and TeleManagement Forum Business Processes. Such a profile would necessitate the alignment of the Building Block Contract with the Contract requirements of TeleManagement Forum, many features of which MODD already supports. Such a TM Forum Profile for MODD would also need to ensure that boundary or external information models used by Building Block Contracts were conformant to the TeleManagement Forum Shared Information & Data (SID) Model. The MODD methodology already uses the same UML notations for such information models, but currently does not restrict the Building Block Contracts to elements of the SID.

A final area in which further research could be achieved is in the automation of the guidelines themselves. An interesting approach would be to use ontologies to represent the guideline design activities, workflows and artefacts and to provide semi automation of each of the guidelines to assist developers in rapidly following them and thus providing improved assistance during the development processes.

References

[Abrahamsson 2002] Pekka Abrahammson, Outi Salo, Jussi Ronkainen, Juhani Warsta "Agile Software development methods – Review and Analysis", ESPOO 2002, VTT Publications 478, ISBN 951-38-6009-4

[Ambler 2006] Scott Ambler "The Agile Unified Process" Available at: http://www.ambysoft.com/unifiedprocess/agileUP.html (last accessed April 30th 2006)

[Adams 1996] E Adams, K Willetts, "The Lean Communications Provider – Surviving the Shakeout through Service Management Excellence", McGraw Hill, ISBN 007070306-X, 1996

[AgileAliance] The Agile Alliance. Website available at: http://www.agilealliance.org/ (last accessed April 30th 2006)

[Beck 2001] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, 'Manifesto for Agile Software Development' available at: http://agilemanifesto.org/ (last accessed April 20th 2006)

[Bezivin 2003] Jean Bezivin "MDA: From hype to Hope and Reality", Keynote Address, Sixth International Conference on the Unified Modeling Language, UML 2003, October 20-24, 2003, San Francisco, California, USA Available at: http://www.sciences.univ-nantes.fr/info/perso/permanents/bezivin/UML.2003/ (last accessed April 20th 2006)

[Booch 1993] Booch, Grady "Object-Oriented Analysis and Design with Applications", Addisson Wesley. ISBN 0805353402 1993

[Calo 2005] Seraphin Calo, Roberto Kung, 'Managing New Networked Worlds', Proceedings of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM'2005), IEEE Communications Society, pages 5-10, 2005

[Caruso 2006] Caruso, "Where is SOA taking us in OSS Design?", Panel Presentation IEEE/IFIP Network Operations Managememt Symposium (NOMS) Canada 2006. Available at: http://www.noms2006.org/ (last accessed April 20th 2006)

[Casewise 2006] Casewise Ltd. www.casewise.com (accessed March 31st 2006)

[Chapman 1995] Chapman, Martin, "Overall Concepts and Principles of TINA", Feb 1995, TB_MDC.018_1.0_94 available at: http://www.tinac.com/specifications/documents/overall.pdf, (last accessed march 31st 2006)

[Coleman 1993] D Coleman 'Object-Oriented Development: The Fusion Method', Prentice Hall, 1993 ISBN: 0133388239

[CTG 1998] "A Survey of System Development Process Models: Models for Action Project: Developing Practical Approaches to Electronic Records Management and Preservation", Centre for Technology in Government, 1998 CTG.MFA 003

[Das 1997] Das, S "ORBWork: A Reliable Distributed CORBA-Based Workflow Enactment System for METEOR2", Technical Report #UGA-CA-TR-97-001, Department of Computer Science, University of Georgia, Feb. 1997

[DMTF] Distributed Management Task Force Inc. Website available at: www.dmtf.org. (last accessed April 20th 2006)

[DMTF 1999] Distributed Management Task Force Inc. "Common Information Model Specification v2.11", available at http://www.dmtf.org/standards/cim/cim_schema_v211/, (last accessed March 31st 2006)

[FCC 2005] Federal Communications Commission "Trends in Telephone services, May 2004. Available at http://www.fcc.gov/Bureaus/Common_Carrier/Reports/FCC-State_Link/IAD/trend504.pdf (last accessed April 20th 2006)

[FlowThru] FlowThru ACTS Project AC335, 'Co-operative secure management of multi technology and administrative domain Network and Service Management Systems' available at: http://www.cs.ucl.ac.uk/research/flowthru/ (last accessed 20th April 2006)

[FORM] FORM official WebSite. https://kdeg.cs.tcd.ie:444/form/ (last accessed March 24th 2006)

[FORM 2002] FORM IST-1999-10357/WIT/WP3/1019-v1.4, Deliverable 9 FORM Project Report; Open Development Framework. Available at: https://kdeg.cs.tcd.ie:444/form/ (last accessed March 24th 2006)

[FORM 2002a] FORM Guideline Documentation Available at: https://kdeg.cs.tcd.ie:444/form/results/deliverables/form_d12.pdf (last accessed April 30th 2002)

[FORM 2002b] FORM Final Inter-Enterprise Management System Model (contains Building Block and Business Process system Specifications) Available at: https://kdeg.cs.tcd.ie:444/form/results/deliverables/form_d11.pdf (last accessed April 30th 2002)

[FORM 2002c] FORM Validation of Inter-Enterprise Management framework (Trial 2) Available at: https://kdeg.cs.tcd.ie:444/form/deliverables.htm (last accessed April 30th 2002)

[FORM 2002d] FORM Validation of Inter-Enterprise Management Framework (Trial 2) – Annex A Operational Requirements Available at https://kdeg.cs.tcd.ie:444/form/results/deliverables/form_d10annexa.pdf (last accessed April 30th 2002)

[Fowler 1997] M Fowler, K Scott, UML Distilled, Applying the Standard object Modelling language, published by Addisson Wesley 1997

[Gallerd 2001] Helena Gallerd, "Ericsson Profile of RUP", Technical Report, 2001 (available at http://www.it.uu.se/research/project/dus/ERUP.pdf last accessed April 20th 2006)

[GB909 2001] TeleManagement Forum "Generic Requirements for Telecommunications Management Building Blocks, part I of the Technology Integration Map Version 3.0" Document number GB909, 2001Available at: http://www.tmforum.org (last accessed April 2004)

[Goldman 2004] Larry Goldman, 'Overview of OSS Market 2004', TeleManagement World, October 2004 also published in 'OSS Market Share', OSS Observer Technical Report, available at www.ossobserver.com (last accessed March 30th 2006)

[Graham 2000] Ian Graham "Object Oriented Methods, Principas and Practices", 3rd Edition, Addisson Wesley 2000 ISBN 020161913X

[Hashimi 2003] Sayed Hashimi, "Service-Oriented Architecture Explained", O Reilly Media 2003. Available at http://www.ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html (last accessed April 20th 2006)

[Hesse 2003] W. Hesse: "Dinosaur Meets Archaeopteryx? or: Is there an Alternative for Rational's Unified Process?" Software and Systems Modeling (SoSyM) Vol. 2. No. 4, pp. 240-247 (2003)

[IETF] Internet Engineering Task Force. Web site www.ietf.org . (Last accessed April 30th 2006)

[IPDR2002] IPDR "Network Data Managent – Usage for IP-Based Services. Service Specification – DOCSIS 1.1 Service Flow Metering", version 3.1 September 2002, Available at: www.ipdr.com (last accessed April 20th 2006)

[ISO 1997] ISO/IEC. *RM-ODP*. Reference Model for Open Distributed Processing. Geneva, Switzerland, 1997. International Standard ISO/IEC 10746-1 to 10746-4, ITU-T Recommendations X.901 to X.904.

[ITU-T 2000] International Telecommunications Union, Recommendation M3000 "Overview of Telecommunications Management Network Recommendations", 2000 (available at http://www.itu.int/rec/T-REC-M.3000-200002-I/en last accessed April 20th 2006)

[Jacobson 1992] Jacobson I, "Object-Oriented Software Engineering: A Use Case Driven Approach" Addison-Wesley 1992, ISBN 0201544350

[Jacobson 1997] Jacobson Ivar, Griss Martin, Jonsson Patrik "Software Reuse: Achitecture, Process and Organization for Business Success", Addisson Wesley, ISBN 0201924765

[Jacobson 1999] The Unified Software Development Process, I Jacobson, G Booch, J Rumbaugh, Addison Wesley, ISBN 0-201-57169-2

[Jacobson 2005] Ivar Jacobson, “A resounding Yes to agile processes -- but also to more” Journal of Information Technology Management, Cutter IT Journal pages Vol 15 No 1 Jan 2005 18-24:

[Jones 2005] Steve Jones, “Toward an acceptable definition of Service”, IEEE Software, Volume 22, Issue 3, May-June 2005 Page(s):87 - 93

[Kande 1998] Mohamed Mancona Kandé, Shahrzade Mazaher, Ognjen Prnjat, Lionel Sacks Marcus Wittig “Applying UML to Design an Inter-Domain Service Management Application” Unified Modeling Language, UML’98 - First International Workshop, Mulhouse, France, June 1998, pages 173–182,

[Kruchten 2000] Kruchten Philippe, ”The Rational Unified Process – an Introduction”,2nd edition, , Addison-Wesley ISBN 0-201-70710-1

[Larman 2003] Craig Larman, V Basili “Iterative and Incremental Development: A Brief History”, IEEE Computer Volume 36, Issue 6, June 2003 Page(s):47 - 56

[Lewis 1999] D. Lewis “A Development Framework for Open Service Management Systems”, Journal of Interoperable Communication Networks, vol. 2/1, pp11-30, Baltzer Science Publishers, Mar 1999

[Lewis 1999a] Lewis, D.; Wade, V.; Bracht, R.; “The development of integrated inter and intra domain management services” Integrated Network Management, 1999. Distributed Management for the Networked Millennium. Also published in Proceedings of the Sixth IFIP/IEEE International Symposium on 24-28 May 1999 Page(s):279 – 292

[Nesbitt 1998] Nesbitt, F., Counihan, T., Hickie, J., The EURESCOM P.610 Project: Providing a Framework, Architecture and Methodology for Multimedia Service Management, Proceeding of 5th International conference on Intelligence in Service and Networks, Antwerp, Belgium, 1998, Springer-Verlag

[OASIS 2005] OASIS "Web Services Business Process Execution Languages version 2.0 December 2005. Available at: http://www.oasis-open.org/committees/download.php/16024/wsbpel-specification-draft-Dec-22-2005.htm (last accessed APril 20th 2006)

[OMG] Object Management Group, website available at www.omg.org (last accessed April 20th 2006)

[OMG 2003] Object Management Group OMG "Model Driven Architecture Guide version 1.0.1 2003. Available at http://www.omg.org/docs/omg/03-06-01.pdf (last accessed April 20th 2006)

[OMG CORBA] Object Management Group (OMG) "Common Object Request Broker Architecture: Core Specification" version 3.03. March 2004. Available at http://www.omg.org (last accessed April 20th 2006)

[Pasley 2005] James Pasley "How BPEL and SOA are chainging Web Services Development", IEEE Internet Computing Vol 9 Issue 3, May/June 2005 Pages 60-67

[Pavon 1996] Juan Pavón: "Towards Integration of Service and Network Management in TINA". Journal of. Network System. Manage. 4(3): (1996)

[Pras 1999] Aiko Pras "Introduction to TMN", CTIT Technical Report (ISSN 1381-3625), 99-9, April 1999, 22 pages. Available at: http://www.simpleweb.org/nm/research/results/publications/pras/ (last accessed April 30th 2006)

[Prospect] EU ACTS 052 Proejct PROSPECT: A Prospect of Multi - Domain Management in the Expected Open Services Marketplace, available at: http://www.cordis.lu/infowin/acts/rus/projects/ac052.htm, (Last accessed April 20th 2006) Further Project details available at: http://www.cordis.lu/infowin/acts/rus/projects/prtitp.htm (last accessed April 20th 2006)

[Romero 2005] Jose Raul Romero, Antonio Vallecillo "Modelling ODP Computational Viewpoint with UML 2.0" Nineth IEEE International EDOC Enterprise Computing Conference (EDOC'05) pages 169 - 181

[Rumbaugh 1991] Rumbaugh, James, "Object Modelling and Design", Prentice Hall, 1991 ISBN 8120310462

[Schulzrinne 2006] Henning Schulzrinne, "Managing the New Internet", Keynote presentation at IEEE/IFIP Network Operations & Management Symposium (NOMS), IEEE Press, April 2006

[Siegel 2005] Jon Siegel "Why use the Model Driven Architecture to Design and Build Distributed Applications", Proceedings of 27th ACM/IEEE International Conference on Software Engineering (ICSE), ACM Press, May 2005, USA

[Sheth 1997] Amit Sheth, "From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Coordination and Collaboration", University of Georgia, ACM SIGGROUP Bulletin, Vol. 18, No 3, December 1997

[Sheth 1999] Amith Sheth, Wil van der Aalst, Ismailcem B. Arpinar., "Processes Driving Networked Economy", IEEE Concurrency, Vol 7, No. 3, 1999

[Strassner, 2003] J Strassner, J Fleck, J Huang, C Faurer, T Richardson TMF White Paper on "NGOSS and Model Driven Architecture (MDA)", version 1.0, 2003 Available at http://www.tmforum.org (last accessed April 20th 2006)

[Strassner 2005] J Strassner, "Policy Management – Challenges of the Future" Keynote presentation, IEEE 6th International Workshop on Policies for Distributed Systems and Networks (Policy2005) Sweden, June 2005

[Sun EJB] Sun Micorsystems, "Enterprise JavaBeans Specification", version 2.1 November 2003. Available at http://java.sun.com/products/ejb/docs.html (last accessed April 20th 2006)

[SWEBOK 2000] IEEE Software Engineering Coordination Committee "Guide to the Software Engineering Body of Knowledge", Available at: http://www.swebok.org ,. (last accessed April 20th 2006

[Tarumi 1997] H Tarumi, K Kida, Y Ishiguro, K Yoshifu, T Asakura, "WorkWeb Systems - Multi Workflow Management in a Multi Agent System", SIGGROUP 1997 [TeleManagenent Forum] TeleManagement FORUM, www.tmforum.org (last accessed April 20th 2006)

[TINAC] www.tinac.com (last accessed March 31st 2006)

[TeleManagement Forum] TeleManagement Forum Web Site http://www.tmforum.org/browse.asp?catID=730 (last accessed March 31st 2006)

[TeleManagement Forum eTOM 2005] TeleManagement Forum eTom Team “Enhanced Telecom Operations Map (eTOM) The Business Process Framework”, release 6.0 Document GB921, November 2005, TMFORM

[TeleManagement Forum eTOM 2005d] TeleManagement Forum eTom Team “Enhanced Telecom Operations Map (eTOM) The Business Process Framework, Release 6.0, Addendum D: Process Decompositions and Descriptions”, Document GB921D

[TeleManagement Forum NGOSS 2005] TeleManagement Forum NGOSS Team ‘The NGOSS Technology Neutral Architecture, Rel 6.0, Document TMF053, nov 2005

[TeleManagement Forum Lifecycle 2006] The NGOSS Lifecycle and Methodology, Rel 4.5 Document GB927, Nov 2004

[TeleManagement Forum SID 2005] TeleManagement Forum “Shared Information/Data (SID) Model Business View Concepts, Principles, and Domains” Release 6.1 Document No. GB922, 2005 Available at www.tmforum.org (last accessed Apirl 30th 2006)

[Thomas 2004] Dave Thomas “MDA Revenge of the Modellers or UML Utopia?”, IEEE Software, May/June 2004 (pages 15-17)

[Twardes 2006] Kevin Twardes “Applying SOA to OSS for Telecommunications”, Panel Presentation, IEEE/IFIP Network Operations & Management Symposium, April NOMS 2006 Available at: http://www.noms2006.org/ (last accessed April 29th 2006)

[Wade 1997] Wade Vincent, Lewis David, Sheppard Mark, Tschichholz Michael & Hall Jane “A Methodology for Developing Integrated Multi-domain Service Management Systems”, 4th International Conference on Intelligence in Services and Networks, LNCS 1238, Springer Verlag, pages 245-254, May 1997

[Wade-1997a] V. Wade, D. Lewis, W. Donnelly, D. Ranc, N. Karatzas, M. Wittig, S. Rao, “A design process for the development of muli domain service management systems, published , Baltzer Science, June 1997 Pages 59-71

[Wade 1998] V. Wade, D. Lewis, W. Donnelly, D. Ranc, N. Karatzas, A Design Process for the Development of Multi Domain Service Management Systems, Guidelines for ATM deployment and interoperability, S. Rao (editor), pages 88-103, Baltzer Science Publishers 1998

[Wade 1999a] Wade V, Muldowney S, Fuller J, “Flexible Automated Enactment of Process Driven Telecommunication Management”, accepted for publication in ‘Interoperable Communications Networks Journal’ - Advanced Strategies and Technologies for Broadband Telecommunications Management’, Vol 1/2, Baltzer Scientific Publishers Neitherlands ISSN 1385 9501

[Wade 1999] V Wade, D Lewis, ‘Three Keys to Developing and Integrating Telecommunications Service Management Systems”, IEEE Network,

[Wade 1999b] V Wade, “Service Management and the Telecommunications Information Networking Architecture”, Computer Communications Journal, Elsevier, Vol 22, Number 18, Dec 1999, ISSN 0140-3664, pages 1633-1638

[Wade 2000] "Workflow, A Unifying Technology for Operational Support Systems", V Wade, T Richardson, IEEE International Symposium on Networks and Operations Management Systems NOMS, Hawaii, USA 2000

[Wade 2002] Vincent Wade, David Lewis, Jacque Brooke, William Donnelly 'Towards a Framework for Management Business-to-Business eCommerce Chain", Chapter VI in 'Challenges Managing Virtual Web Organisations in the 21st Century: Issues and Challenges' Ed. Ulrich Franke, pages 107-118, published by Idea Group Publishing, 2002

[Wade 2002a] Vincent Wade, David Lewis, "A Model-Driven Approach to Component Based Management, Tutorial T9, IEEE 8th International Symposium on Networks and Operations Management Systems NOMS, Florence, Italy, April 2002

[Wade 2002b] V Wade, D Lewis , IST-1999-10357/TCD/WP3/012D12 – FORM Methodology for the Open Development Framework. Available at: https://kdeg.cs.tcd.ie:444/form/results/deliverables/form_d12.pdf (last accessed April 30th 2006)

[Wagner 2003] Stefan Wagner "NIMSAD Evaluation of the Rational Unified Process", Available at http://www.stefan-wagner.info/cs/nimsad.php (Last accessed APrili 30th 2006)

[WC3 2004] W3C "Web Services Architecture", Available at: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/ (last accessed April 20th 2006)

[W3C 2005] W3C "Web Services Choreography Description Language Version 1.0 (W3C Candidate Recommendation)", November 2005. Available at http://www.w3.org/TR/2005/CR-ws-cdl-10-20051109/ (last accessed April 30th 2006)

[Weissenfels 2006] J Weissenfels, D Wodtke, G Weikum, A Kotz Dittrich, http://paris.cs.uni-sb.de/public_html/papers/mentor.html (Wikapedia 2006).

[Westerinen 2001] Andrea Westerinen, "CIM, The Common Information Model Tutorial", IFIP/IEEE International Symposium on Integrated Management (IM2001), 2001

[WfmC 2001] Workflow Management Coalition 'Workflow Handbook, 2001 ed. Layna Fischer, Wiley ISBN 0471969478 2001

[WSDL 2001] W3C "Web Srevices Definitio Language", V 1.1 March 2001 available at: http://www.w3.org/TR/wsdl (last accessed 20 April 2006)

[Yates 1997] Martin Yates, Wataru Takita, Laurence Demoudem, Rickard Jansson, Harm Mulder, "TINA Business Model and Reference Points", Version 4, May 1997 (Available at www.tinac.com/specifications/documents/bm_rp.pdf) March 31st 2006-04-02

Glossary

Artefact — An artefact is a piece of information that is created, changed and used by actors when performing development activities. An artefact can be a model, a model element or a document.

Boundary Information Model: This is the information model, which is communicable across a Building Block Contract. This Information Model is therefore the information which is externalised by the Building Block Contract and which is available (either as input or output) to any user of the Building Block Contract.

Building Block — A Building Block (BB) is an atomic unit of software deployment and software management. A BB implements a number of Contracts that are the sole medium for inter-BBs interactions.

Building Block Contract — Building Block Contract is an interoperability specification which provide the only means of interacting with a BB. A Building Block Contract specification contains a grouping of interface signatures, information models and interaction behaviours, which can be re-used to support telecommunication management business processes.

Building Block Development Guideline: The Building Block Development guideline describes a *development process, which facilitates the development of Building Blocks and Building Block Contracts.*

Business Process: This is a workflow, which describes (telecommunication or application) business activities, which have to be performed to achieve a business goal.

Management System Process: This is a workflow, which describes management activities, which have to be performed to achieve a management goal. Typically this term is used to identify a sequence set of management activities, which are to be supported by computerisation.

(Telecommunication) Management Business Process: This is a workflow, which describes (telecommunication or application) management activities, which have to be performed to achieve a management goal. Typically this term is used to identify a sequenced set of management activities, some of which are to be supported by the Building Blocks in the Open Development Framework.

Package: A general-purpose mechanism for organising elements into groups. Packages may be nested within other packages. All kinds of model elements and diagrams can be organised into packages. Special kinds of packages are model and system.

Phase (or Development Phase): A Phase represents the time between two major project milestones during which a well-defined set of objectives is met, artefacts are completed, and decisions are made to move into the next phase. Each phase has a set of workflows, which determine and sequence development activities. Each Phase may iterate these workflows several times to allow artefacts to be refined.

Reference Point: A Reference Point is the relationship, which is modelled at the Business level between business management processes, which reside in different Business Roles. They indicate a level of interaction typically between two organisational units and their respective processes. This is important in modelling B2B domains.

Subsystem: A subsystem is a grouping of classes or other subsystems. Subsystems can be devised either bottom-up or top-down. When working bottom-up, subsystems are suggested based on classes already found. Working top-down means that high-level subsystems and their interfaces are identified before the classes are identified.

Workflow: A workflow identifies a set of activities and their sequencing to achieve a particular goal.

Workflow Process: This consists of the sequence development activities required to achieve a particular development goal. It can also consist of other Workflow Processes

Appendix 1: Relationship between the Building Block Guideline and Rational Unified Process

Technical Approach for Building Block Development Guideline

The approach taken in developing the Building Block Development Guideline was to re-use current best practice in software development and to customise and add features or artefacts where required. Therefore the Guideline was not devised from scratch, but rather constructed from the most widely accepted methodologies and then enhanced to suit its needs. Several candidate software development processes were identified [SWEBOK2000], but it was decided to base the 'Context Modelling' workflows loosely on Rational's Unified Process (RUP). Several reasons underpinned this choice:

(i) RUP is widely adopted in the Object Oriented software development community.

(ii) RUP employs the Unified Modelling Language, which is a modelling notation adopted widely in industry and by many standardization fora in relevant areas such as management, e-commerce and distributed computing

(iii) RUP is a 'development process framework' and thus it is intended to be customised for the development of different specialised artefacts and processes. It provides the flexibility needed to adapt RUP to the particular methodological requirements of the ODF stakeholders

(iv) RUP claims to support component oriented as well as Object Oriented software development.

Therefore some of the Building Block Guideline workflows are a customisation and enhancement of known industrial software methodologies. However it has been augmented with best practices based on experiences in previous research projects and academic work. The Building Block Development Guideline focuses on the Development of Building Blocks rather than the mapping of such models into specific technologies or computing platforms. It also facilitates the specification of Building

Block Contracts. A Building Block Contract specifies a grouping of information and behaviours, which can be re-used to support management business processes. A Building Block Contract can be supported by one or more Building Blocks.

Rational Unified Process (RUP)

The Rational Unified Process (RUP) is a software engineering process developed and marketed as a product by Rational Software. RUP is itself a specialisation of the Unified Software Development Process (USDP). RUP is delivered online using Web based technology and consists of more than 1000 hyperlinked pages of text and graphics. It provides a proven disciplined (industrial) process for assigning tasks and responsibilities within a development organisation to design applications and enterprise systems. RUP aims to capture many of the best practices in software development and then attempts to present them in a form that can be tailored for a wide range of projects.

This section briefly identifies the important aspects of RUP, which are utilized later in describing the Building Block Development Guideline. A more comprehensive overview of important aspects of RUP is contained in [Kruchten 2000] and a broader description of UML based development process is also presented in [Jacobson 99].

RUP depicts software development in two dimensions, Phases (Inception, Elaboration, Construction Transition) and Process Workflows (i.e. development activities), which are conducted within each phase. RUP also identifies three Supporting Workflows, which support the co-ordination of the overall development effort called Change/Configuration Management, Management and Environment). Figure 1 represents the four phases, the process workflows and also provides an indication of the level of effort devoted to each process work within each phase.

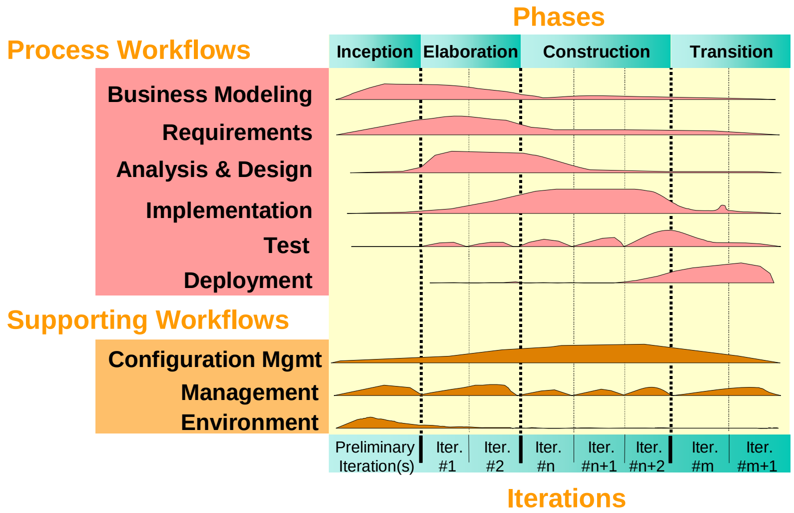


Figure 1 Rational Unified Process Lifecycle Model

RUP is an iterative process in that multiple iterations of the process workflows are expected within each phase. The precise number of such iterations is dependent on the complexity of the solution being developed and the operating context of the development effort e.g. the experience of developers, complexity of application area etc.

The goal of the Inception Phase in RUP is to develop the business case to the extent necessary to justify launching the project [Jacobson2000]. In this phase the workflows determine the scope of the system to be developed as well as developing parts of the models that would be necessary to support a proof of concept prototype. However the business modelling, requirements and analysis workflows are the principle areas of effort, whereas the implementation, test and deployment workflows concentrate on planning activities and infrastructure selection.

The Elaboration Phase has several specific targets, namely the capturing of 80% of the required use cases of the system under investigation, commencement of detailed design work, completion of a deployment model for the envisaged system, and completion of about 10% of the implementation work completed. The Construction Phase should achieve a complete system implementation ready to begin transition to a user community.

The Transition Phase involves the deployment of the completed system into its intended user community and the performance of minor fixes and some fine-tuning.

Therefore, the Phases in RUP define ‘When’ in the software development lifecycle activities should be performed. The ‘Who’, ‘What’ and ‘How’ of the development process are defined within the Process and Support Workflows. Thus, in each of the workflows, workers or roles indicating who should carry out a development activities are identified, how development activities should be performed and what artefacts (models, design elements) are required to be developed. The Process Workflows defined in RUP are:

(i) Business Modelling,
(ii) Requirement Management,
(iii) Analysis & Design,
(iv) Test,
(v) Deployment

and the Supporting Processes are:

(iv) Change/Configuration Management,
(v) Project Management and
(vi) Environment.

Relationship between the BB Development Guideline and Rational Unified Process

Implementing a system based on RUP can be achieved by authoring and adhering to a development case, which defines the specialisations/additions to the original RUP phases, activities, process workflows and artefacts. The Building Block Development Guideline can be thought of as such a development case, which facilitates the development and specification of Building Blocks and Building Block Contracts. In particular the development case focuses on tailoring the various aspects of the RUP’s business modelling, requirements modelling, analysis & design, implementation and deployment workflows.

The Context Modelling Phase of the BB Development Guideline can be thought of as a filter on RUP that selects and customises the activities that are to be performed and artefacts that are to be produced as well as adding additional elements where necessary. For the Open Development Framework, it was found that RUP was lacking in support for a "component design". In RUP, a component is a runtime unit of software, whereas in FORM the desire is to provide a reusable grouping of models and artefacts, which is rich enough to adequately represent the properties, pre- and post- conditions, constraints, potential usage, and interfaces of a Building Block. This grouping is termed a Building Block Contract. A Building Block may support one or more Building Block Contracts. A FORM Building Block more closely resembles a "Sub System" in RUP.

This Guideline customises the RUP phases to reflect the goal of developing Building Blocks and Building Block Contracts for the FORM Open Development Framework.

Mapping the Guideline Development workflows into an extended RUP

The Building Block Development Guideline can be presented as an extension and customization of several aspects of RUP. RUP was designed for a broad range of application and enterprise systems. However, this guideline is not focused on application development, but rather reusable component development. Therefore, augmentation and extension of various parts of RUP were necessary because of the development target. For example, because the guideline is focused on the development of management components, considerable effort is needed to initially model the problem domain in order to identify and scope candidate components as well as to design and specify the components (once they are agreed). Also, management systems (for service and network management) are frequently distributed across organisational or administrative boundaries. Such management system boundaries are usually represented in the telecoms area as reference points[1]. Such reference points then become sets of coarse-grained integration points for (possibly multi-vendor) software components. Such reference point

[1] Different standards bodies have used the notion of reference points to assist in describing relationships between either business entities or systems e.g. TINA.

specifications are not commonplace in other software markets and domains and hence are not currently part of RUP. The customisations of RUP included changes to Development Phases, Model & Artefact Selection, Process Workflows and Documentation.

Development Phases & Iterations.

The Context Modelling Phase for this Guideline is more extensive than would normally be the case for general application of the Inception Phase of RUP.. The Business Modelling work is very important in initially scoping the management process areas and domain responsibilities for which the Building Blocks will provide support. The Business Modelling work involves identifying the domain boundaries of the envisaged management systems and the organisation in which they may reside, identifying management use cases and initially modelling the management processes which support them. An additional outcome of this phase is the modelling of analysis objects, which support these use cases and management processes. These analysis objects are used in the Building Block Development Phase in the identification and determination of candidate Building Blocks. Also this phase allows the modelling of reference points between Business Organisations identified during business modelling. External influences in this phase include the adherence/alignment with relevant telecom industry standards e.g. TeleManagement Forum’s Telecom Operation Map (and Fulfilment-Assurance-Billing processes) [TeleManagement Forum]. Such standards can have a significant influence on the management processes being modelled as well as on the information (object) model(s) associated with these processes. However, it is important to remember that the objective of this development Guideline is the development of Building Blocks and Building Block Contracts. The effort therefore in this (inception) phase is to identify the business processes, actors, potential organisational boundaries, etc. which impact on and place in context the building blocks and building block contracts which will be subsequently developed in the later development phases.

The Building Block Development Phase focuses on the development of specific building blocks and Building Block Contracts and involves testing of these Building Blocks. This involves the identification of Building Blocks and the re-organisation and grouping of analysis objects into re-usable Building Block designs and specifications. The elaboration phase also involves the modelling of information used both within the building block(s) as well as those (information objects) passed into or communicatedout of the Building Blocks. The information communicated by a Building Block Contract is termed its 'Boundary Information Model' and is documented as an explicit information model indicating the informational requirements and outputs of the Building Block. The key output of this phase is the specification of Building Block Contracts and the design of Building Blocks, which support them.

The Construction Phase and Transition Phase is not considered within the BB Development Guideline.

Models and Artefacts Selection.

RUP defines a very large set of potential modelling and documentation artefacts. The BB Development Guideline prescribes a reduced set of models necessary to design and specify Building Blocks. Also the guideline specifies a Building Block Specification Template, which presents a rich description of the Building Block, essential for later reuse.

Two additional artefacts, not in RUP, are contained in the Guideline, namely:

(i) An explicit architectural (structural) diagram, called the Reference Architecture, which indicates reference points (boundaries) between management/administrative boundaries and the placement of business processes within and/or across these boundaries. This architectural diagram attempts to reconcile the management process areas (for which building blocks are to be developed) with the organisation or administrative boundaries being considered within the Business context.

(ii) The explicit modelling of Building Blocks and specification in XML of the Building Block Contracts. These Building Block models are considered fundamental to the FORM Framework and do not have a direct equivalent in RUP. The most similar package of models defined in RUP, which includes the grouping of such required models, is that of a Sub System package.

Process Workflows

The Guideline concentrates on the Process Workflows (rather than the support workflows). This does not mean such support workflows are not relevant, on the contrary they are very useful. However, as this guideline is focused on the actual development activities (rather than their management or product deployment), the Process Workflows are the only activities customised. This customisation is necessary for the development of Building Blocks and the modelling artefacts that are used to specify them.

Documentation

The Guideline focuses on only a few key document templates, including Vision Document, Software Architecture and Building Block Specification.

Terminology used in the Guideline

The concepts and terminology typically used to describe methodologies for software design can be confusing. This section provides some simple definitions of the various terms used by the Guideline. The Building Block Development guideline describes a *development process*. A development process consists of a *set of phases,* which provide major checkpoints during the development. Each phase has associated with it a set of *workflow processes,* which determine and sequence *development activities.* Each Phase may iterate these workflow processes several times to allow the design models to be refined. Later Phases expend much greater effort on software implementation issues whereas implementation issues, when addressed in earlier phases, tend to focus on infrastructure planning and proof of concept. The Glossary contains complete definitions of key terms used in the Guideline.

APPENDIX 2: Presentation of MODDs guideline with example of Building Block Specifications and Business Process Driven Systems

The following slides were presented as a four hour tutorial at the IEEE/IFIP Network Operations Management Symposium 2002. The majority of the presentation focuses on the MODD development guideline and was performed by the author. These guidelines were developed and presented by the author. The other parts of the tutorial were presented by D Lewis (who presented context for the use of MODD), B Lonvig and B Bhushan. The latter two presenters were two of the FORM developers who applied MODD in the design and implementation of example Building Blocks (Contracts) and business process based Management systems. Their slides indicate how they applied MODD in two case studies.

A Model Driven Approach to Component Based Management

Vincent P. Wade
Trinity College Dublin
Vincent.Wade@cs.tcd.ie

David Lewis
University College London
D.Lewis@cs.ucl.ac.uk

Birgitte Lønvig
Ericsson Denmark
Birgitte.Lonvig@ericsson.dk

Bharat Bhushan
Fraunhofer FOKUS
bhushan@fokus.fhg.de

Difficulties in Management System Development Today

- Typically integrate separately-sourced systems within an operators network
  - e.g. element agents, network management consoles etc
- Impact of (sometimes) contradictory industry standards e.g. TMF, OMG, IETF, DMTF
- Successive waves of new technology hype and roll-out make stable software architectures difficult to achieve

Difficulties in Management System Development Today

- Typically large and inevitably use multiple technologies which must interwork
- Move toward third party components means system developers loose full control of software architecture
- Need to understand a minimum set of shared, long-lived architectural concepts

Motivation for This Tutorial

- System developers must deal with:
  - Multiple sources of component software
  - Multiple sources of interoperability standards
  - Multiple technologies
  - *.. yet protect investment in business logic design*

Motivation for This Tutorial

- Propose a Development Methodology aiming to help *all the players involved* in management software development: Independent S/w vendors, System Integrators, Service Providers, Standards Bodies
- Benefits of Methodology:
  - Protect Analysis &Design investment
  - Reduce cost of technology 'churn'
  - Communicate and re-use designs more easily
  - Focused development for each 'player'
  - Apply & evolve best practice is component & system development

Overall Tutorial Goal

To examine and demonstrate model-based approaches for development of component based management systems

Specific Objectives

- ***Identify motivations & trends*** in model-based approach to management system development
- ***Survey state of the art*** in modelling systems
- Identify and iterate ***modelling guidelines*** the development of ***Building Blocks***
- Identify and iterate ***modelling guidelines for*** the development of ***management systems***
- Present and Demonstrate ***case studies***

Tutorial Outline

Section I: **Motivations and Trends in Component Based Development and Model Driven Development**

Section II: **Development Approaches for model-driven, component-based management systems development**
- Overview, Developing Management Components, Accounting Case Study

COFFEE BREAK

Section III: **Development Approach Part II: Business Process Driven Systems Development**
- System Development Guideline
- Case Study:Fulfilment Business Process

Section IV **Lesson Learnt and Future Developments**

DEMO

Section I
Motivations and Trends in Component Based Development and Model Driven Development

David Lewis
University College London

The Evolution of Telecom Management Technologies

- 1980s: Management specific protocols
  - SNMP, CMIP
- 1990s: General purpose distributed computing
  - DCE (DME), CORBA, COM, RMI, …
- Today:
  - Components-based Software
    - EJB, .NET, COBRA Components
  - Web-enabled management
    - WBEM, SOAP, WSDL …

Technological (Un)Certainties

- There is a steady flow of technological innovation applicable to management systems
- Many innovations have involved adoption of different technology specific models and interaction paradigms, e.g.:
  - GDMO and manager-agent paradigm for CMIP
  - IDL and RPC paradigm for CORBA
  - XML and message passing for SOAP
- Developers must acquire new technology skills whilst maintaining and enhancing existing management know-how!
- Will always need technology interworking to accommodate legacy systems

Handling Technological Change

- Management software development needs to address accelerating technological change
- We will examine approaches taken by:
  - Operational Support System through Java (OSS/J)
  - Distributed Management Taskforce (DMTF)
  - TeleManagement Forum's NGOSS

OSS/J

- A set of management APIs
  - Service Activation, billing, Trouble Ticketing
  - Defines common objects and design patterns
- Builds on the J2EE platform
  - EJB component/container model
  - Container services for directory/naming, persistence, transaction, security
  - Container mediated interactions;
    - RPC: RMI, CORBA
    - Message based: JMS, JAXM
    - Connectors to 'legacy' protocols: SNMP, CMIP, TL1

DMTF

- Manager-Agent approach based on the definition of managed objects
- Technology Neutral Common Information Model captured using:
  - UML class diagrams
  - Managed Object Format (MOF)
- Multiple protocol mappings:
  - Desktop Management interface – RPC-based, DCE mapping
  - Directory Enabled Networks – mapping to LDAP/X.500
  - Web-Based Enterprise Management – Mapping to XML/HTTP

NGOSS

- An architectural framework for component-based OSS
- 3-tiered architecture with process flow management
- Components offer contracts which have:
  - Technology neutral and technology specific forms
  - Pre- and post-conditions
- TM Forum aims to standardise:
  - Process Model – eTOM
  - Contracts
  - Shared Information

Approaches to Handling Different Technologies

- OSS/J focuses on software development and integration:
  - Does not address use of models in non-J2EE environment
- DMTF focuses on interoperability models, no component-model
  - Information model style may restrict effective mappings to future
- NGOSS captures abstract Contracts and Shared Information Model derived from Business Process Model
  - Needs to manage technology specific mapping to ensure interoperability

Re-using Components

- Component technologies such as J2EE allow integration of separately sourced components and with legacy systems
- Management system developers may be faced with a wide range of potentially useful 3rd party components and legacy systems
- Effective reuse of existing systems and use of commercial off-the-shelf components requires an understanding of:
  - The relevance (or otherwise) of a component's capabilities to the problem at hand
  - The semantics of the component behaviour

Developing Reusable Components

- Need to have long-lived designs which can map onto the integration technology for a specific application
- Need to protect 'design' investment and ensure ROI
- Need to avoid unnecessary overheads related to 'technology churn' – technology driven remodelling
- Overall - We need a Model Driven Approach to both component development and component reuse

©VW,DL,BB,BL NOMS 2002 18

Methodologies for Model Driven Development

- A wide range of development methodologies have been developed:
  - BOOCH's OOAD, OOSE, Rational Unified Process (RUP), The Open Process …
- No single 'best development process' has emerged
  - Target systems have differing requirements w.r.t., development-time, cost, longevity
  - Culture and skill set of the developers
- The Unified Modelling Language (UML) has emerged as standard graphical modelling notation

©VW,DL,BB,BL NOMS 2002 19

Unified Modelling Language

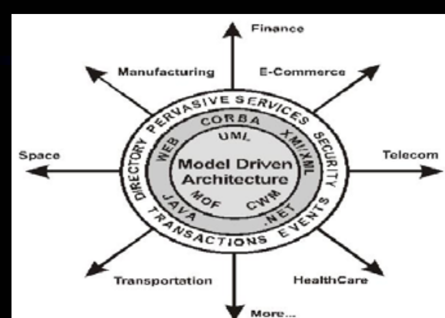
- Moves us towards modelling the whole of software development lifecycle:
  - Requirements Capture
  - Analysis
  - Design
  - Implementation
  - Deployment
- Provides basis for CASE tools
- Needs to guide use when applying UML in a domain-specific process
  - Restrictions
  - Extensions - stereotypes

OMG's Model Driven Architecture (MDA)

- Provides a guide for structuring software specifications expressed as models
- Separation of Platform Independent Model and Platform Specific Models, i.e.
  - Separation of specification of system fuctionality from the specification of its implementation on a specific technology platform
- Core of Approach is UML based (technology independent) Modelling
- Contains:
  - A core meta model for UML model and derived technology specific models (Meta-Object Facility)
  - A mechanism for exchanging models between CASE tools (XML Model Interchange)

©VW,DL,BB,BL NOMS 2002 21

OMG's MDA



©VW,DL,BB,BL NOMS 2002 22

Section II Development Approaches for model-driven, component-based management systems development

Vincent P. Wade
Trinity College Dublin

©VW,DL,BB,BL NOMS 2002 23

Recap: Influences in Management System Development

- Integration separately-sourced systems within an operators network
  - e.g. element agents, network management consoles etc
- Impact of (sometimes) contradictory industry standards e.g. TMF, OMG, IETF, DMTF
- large and inevitably use multiple technologies
- Successive waves of new technology hype and roll-out
- Move toward third party components means system developers loose full control of software architecture
- Need to protect development investment in face of technology churn

A Model Driven Approach to Component-Based Development for the Management Domain

- This must take into account the needs of component developers and component reusers
- Should follow industry best-practice, but support needs of Management Domain, e.g.:
  - Wide range of technologies
  - Specialised interaction paradigms, e.g. Manager-agent
  - Standardisation of models
- Should be practical to enact with available CASE tools
- Should support practical software development in different technologies

Requirements for a Development Methodology

- Communicating designs/products in a mutually understandable form
- Converge with industry best practice in software development, e.g. use of UML
- Support the publication, location and examination of the exchanged wares
- Consider but not prescribe technology selection
- It must encourage the separation of technology interworking from model interworking

A Proposed Solution:

Development Methodology for Component Based Management systems
that utilises Model-Based Development and Open Interfaces:

Component based Model Driven Development Approach

A Model Driven Approach

- Avoids prescribing technologies, integration techniques or software architectures
- Focuses on the generation and management of ***models***
- Exploits the widespread understanding of UML and the publication benefits of XML
- Focuses on different users of the Methodology by offering different Guidelines

Architectural Principles in Brief

- Systems assembled from Building Blocks
- Building Blocks support one or more Contracts
- Contracts support multiple business operations
- Contracts are (ideally) defined in technology neutral form, with mappings to specific technologies
- Information content of Contracts is modeled explicitly to aid reuse
- Business Logic may be modeled separately to Contracts and Building Block to aid flexibility

A Methodology for Model-Driven Development

General Methodology Objectives

- Provide guidance as to design phases and workflows
- Identify notations to be used to describe artifacts
- Identify artifacts to be developed for each phase & workflows
- Direct tasks/workflows for development effort

Focus on the Modelling Aspects of the development work rather than the implementation and technology aspects

Audience for Methodology

- Methodology aimed at supporting contrasting development requirements of the management actors (examples based on B2B management)
  - Management Component Vendors
  - Management System Integrators
  - Inter Enterprise Service Providers
- Methodology Divided into Two Guidelines

  => more focused assistance on modelling tasks and problems

Development Methodology

GUIDELINE #1:

– Guideline for development of re-usable management components (termed Building Blocks)

GUIDELINE #2:

– Guideline for construction of management systems to implement management business processes using (pre-existing) Building Blocks

Guideline #1 Objectives

- Provide guidance as to design activities in developing Building Block(s)
- Identify what notations to be used
- Identify artifacts to be developed
- Direct tasks/workflows of Building Block development effort
- Prescribe combination of artifacts to characterise and communicate usage of Building Block (i.e. re-use)

Building Block Development Guideline

Scope:

– Define development workflows to model BB Contracts and BBs

– Focus on design activities rather than implementation techniques (coding) and testing techniques (e.g. unit testing etc.)

– Attempt to remain technology independent in modelling effort – (however as there is no technology neutral interface specification lanaguage, native interface specifications can be imbedded in BB Contract definitions)

Development Approach

- Use current best practice in software development and to customise elements of methodology where required
- Use of UML as modelling language
- Some development activities loosely based on RUP process workflows
- Use XML to publish Building Block Contracts

Development Approach

- Challenges in Building Block Development:
  – Difficult to initially identify useful BBs in a problem area (e.g. QoS, Accounting, Configuration Mgt)
  – Need to define Building Block & Building Block Contract Specifications and model groupings
- Appoach taken to identify useful BBs is classical top down approach but then apply 'bottom-up' techniques
- Define XML based specifications of Building Block Contracts & descriptions for Building Blocks.

Overview of Building Block Guideline

Divide Guideline into two Plases:

– Context Modeling Phase

– Building Block Development Phase

Principal Workflows: Context Modelling Phase

– Perform Business Modelling

– Refine Reference Architecture

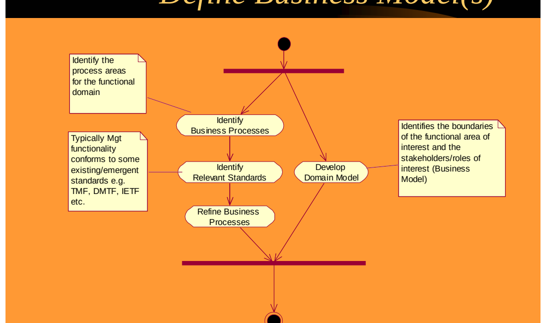
– Perform Requirements Analysis

– Develop Analysis Object Models

Principal Workflows: Building Block Development Phase

- Re-iteration of Previous Phase workflows: Business Modelling, Reference Architecture Modeling, Requirements Analysis, Analysis Object Modeling
- Re-organise Analysis Model & identify candidate BBs and BB Contracts
- Model BB Contracts and provide XML specification
- Develop BB Descriptions in BB Catalogue

Define Business Model(s)

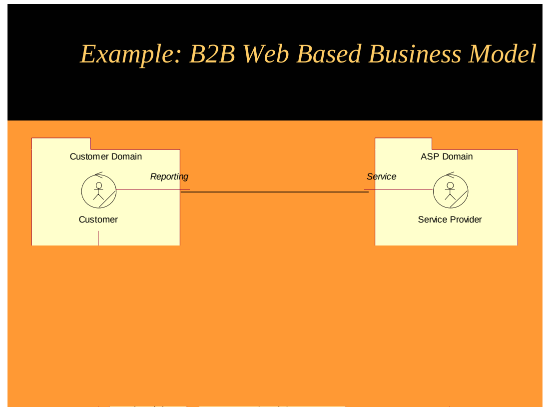


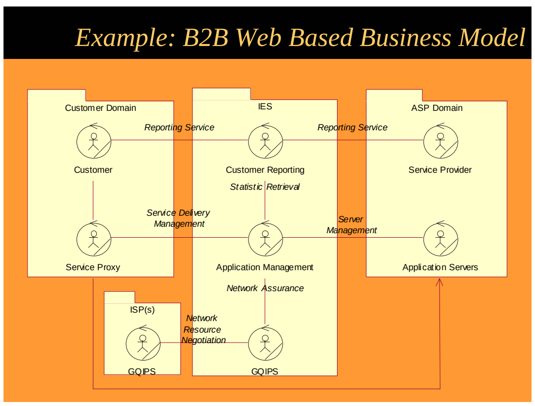
Artifacts Produced by Perform Business Modelling

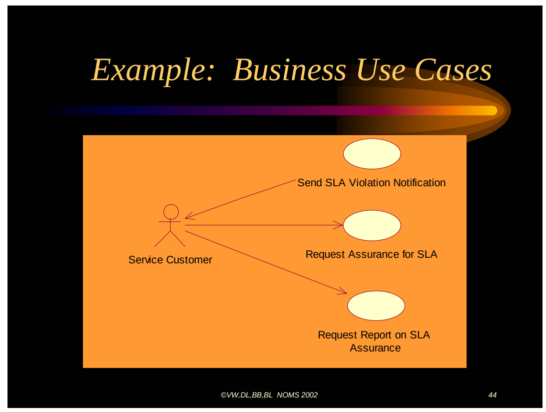
Business Use Case Diagram(s): depicting business roles (workers and/or organisations), use case name (external view)

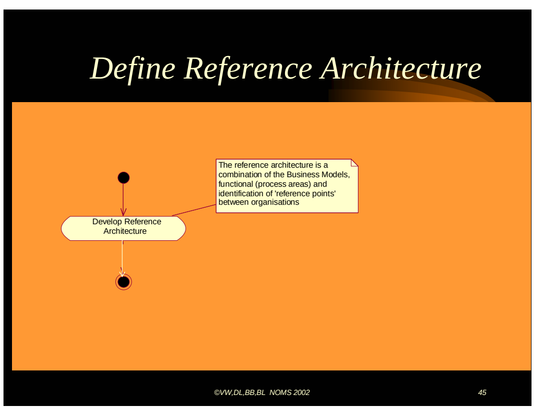
Use Case Realisation, which models the business workers and entities/resources needed to carryout the use case (internal view)

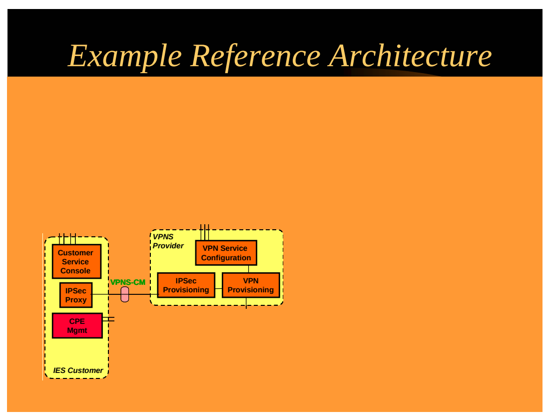
Activity Diagram depicting the activities involved in carrying out the use case (internal view).

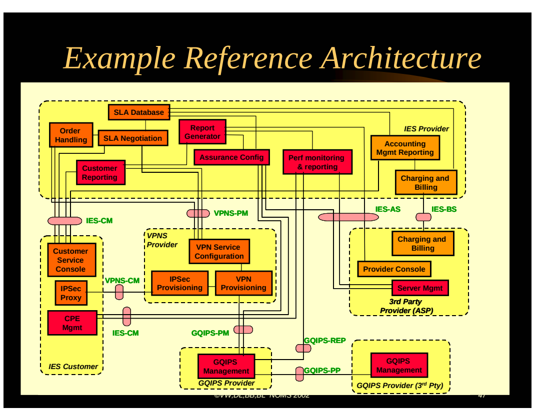


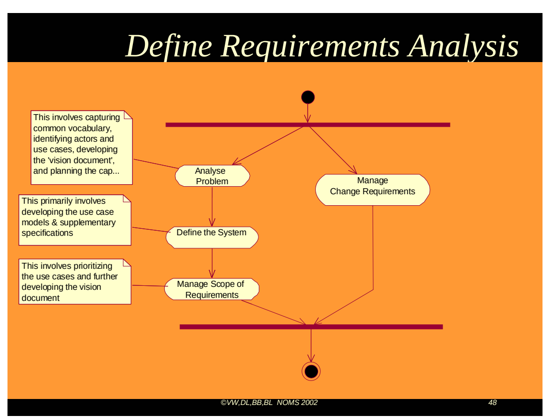


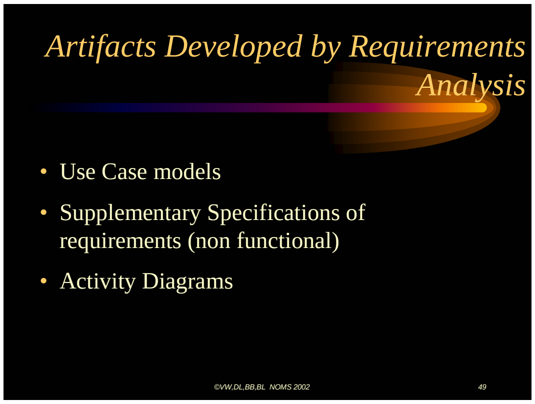


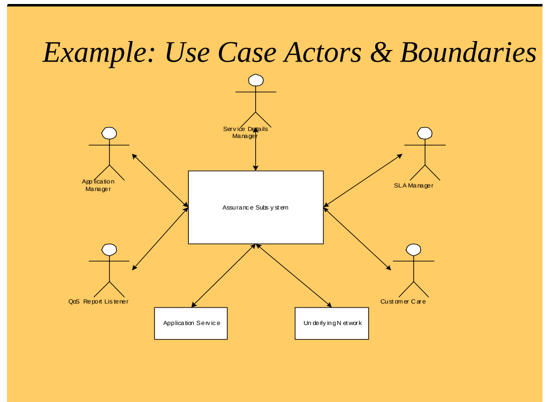


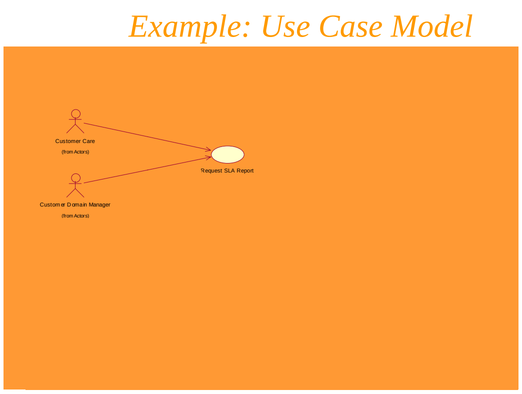


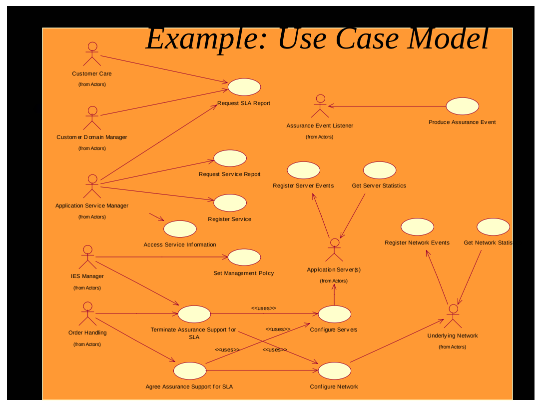


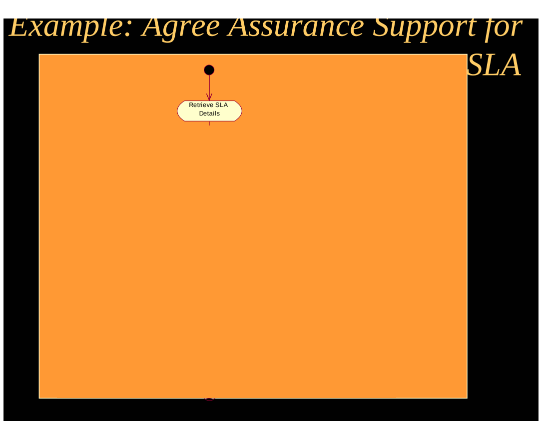


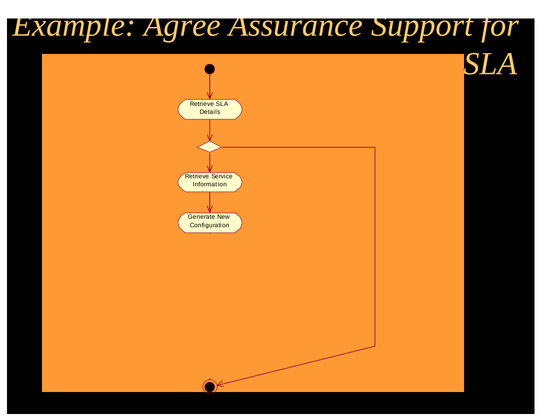

Artifacts Developed by Requirements Analysis
Use Case models
Supplementary Specifications of requirements (non functional)
Activity Diagrams
©VW,DL,BB,BL NOMS 2002
49

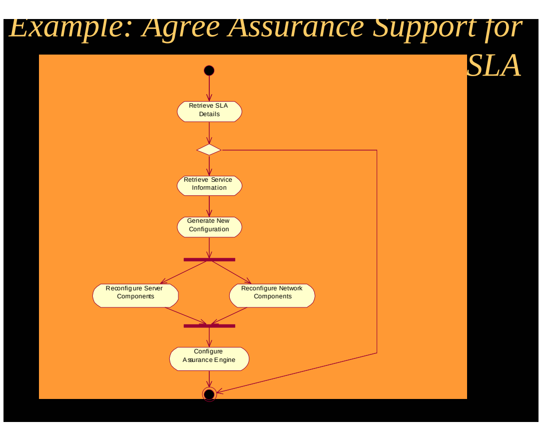

Example: Use Case Actors & Boundaries
Service Details Manager
Application Manager
SLA Manager
Assurance Subsystem
QoS Report Listener
Customer Care
Application Service
Underlying Network

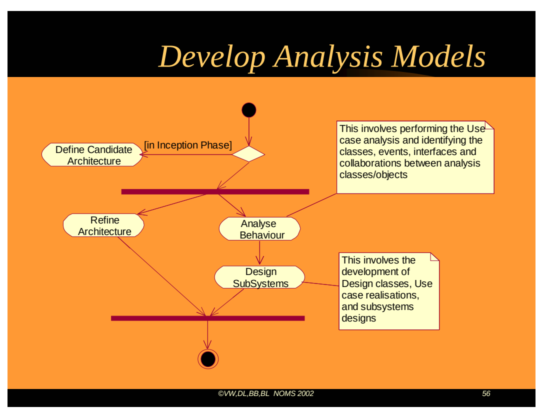

Example: Use Case Model
Customer Care
(from Actors)
Request SLA Report
Customer Domain Manager
(from Actors)


Example: Use Case Model
Customer Care
Request SLA Report
Assurance Event Listener
Produce Assurance Event
Customer Domain Manager
Request Service Report
Register Server Events
Get Server Statistics
Application Service Manager
Register Service
Access Service Information
Register Network Events
Get Network Stats
Set Management Policy
Application Server(s)
IES Manager
Order Handling
Terminate Assurance Support for SLA
<<uses>>
Configure Servers
Underlying Network
Agree Assurance Support for SLA
Configure Network

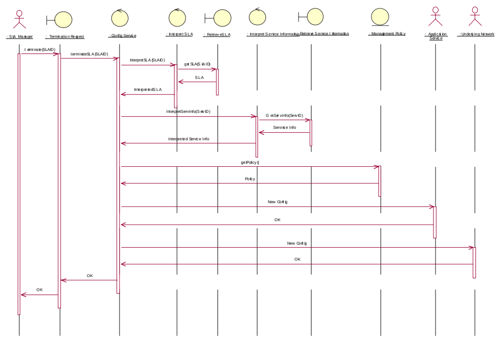

Example: Agree Assurance Support for SLA
Retrieve SLA Details


Example: Agree Assurance Support for SLA
Retrieve SLA Details
Retrieve Service Information
Generate New Configuration

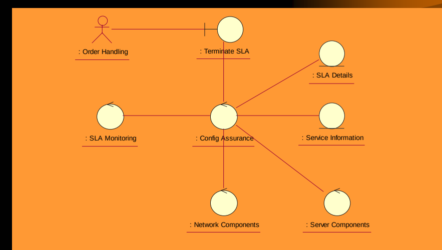

Example: Agree Assurance Support for SLA
Retrieve SLA Details
Retrieve Service Information
Generate New Configuration
Reconfigure Server Components
Reconfigure Network Components
Configure Assurance Engine


Develop Analysis Models
Define Candidate Architecture
[in Inception Phase]
This involves performing the Use case analysis and identifying the classes, events, interfaces and collaborations between analysis classes/objects
Refine Architecture
Analyse Behaviour
Design SubSystems
This involves the development of Design classes, Use case realisations, and subsystems designs
©VW,DL,BB,BL NOMS 2002
56

Example: Interaction Analysis Object Model



Example: Analysis Object Modelling



©VW,DL,BB,BL NOMS 2002 58

Development Workflows of Building Block Development Phase

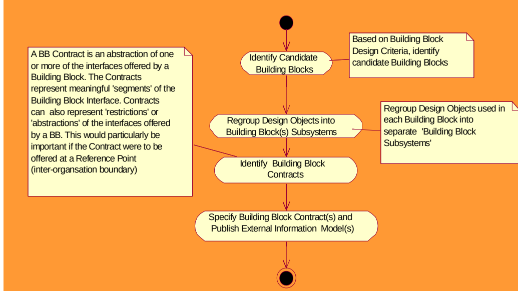
- Revision & refinement of the Business Modelling, reference architecture, requirements management and analysis models
- Reorganisation of analysis model for candidate Building Block groupings.
- Specify Building Block Contracts
- Perform testing of proposed computing and testbed infrastructures.

Main Artifacts Produced:

- A stable system architecture model.
- Initial Development Models for Building Blocks Contracts.

©VW,DL,BB,BL NOMS 2002 59

Re-organise Analysis Model



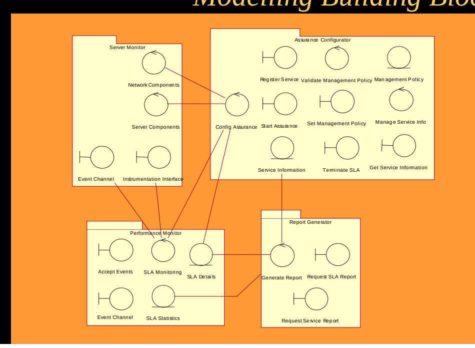
Some Simple Criteria for identifying candidate BBs

Does the grouping of classes provide Enterprise Wide information service, re-usable business logic or generally useful User Interface (i.e. at the Enterprise Information Tier, Process Automation Tier or Human Interaction Tier)?

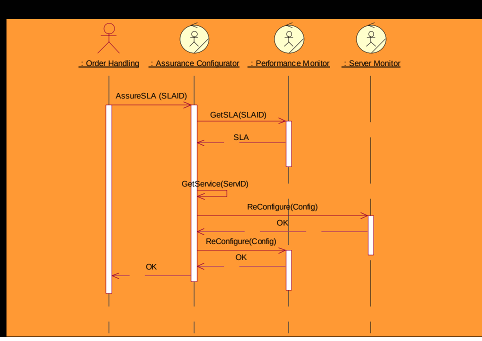
Does the grouping of classes represent some self-contained behaviour (logical grouping of closely related behaviours)?

- Level of inter-dependence between grouping of classes and interacting classes (based on original use cases)?
- Is there a definite 'service' or 'services' the BB can uniquely support (does it add a useful, distinct, service to the system)?

Re-organising Analysis Models for Modelling Building Blocks



Interaction Diagram: Agree Assurance Support for SLA



Specify Candidate Building Block Contracts

- A Building Block Contract Name (specified as text).
- The names of defined Reference Points Supported by the Building Block Contract (if any).
- Contract description defining service offered by Building Block Contact (specified as text).
- Use cases & collaboration diagram(s) to illustrate usage scope of Building Block Contract
- Contract interface specification.
- Information Objects communicated at the interface of the Building Block Contract (Boundary Information Model).
- Technological description for Building Block Contract (specified as text).
- Collaboration diagrams illustrating this Building Block's potential interaction with other Building Block Contracts (optional)

©VW,DL,BB,BL NOMS 2002 64

Case Study:

Model-based Development of Building Blocks for Billing Management System

Bharat Bhushan
Fraunhofer FOKUS
bhushan@fokus.fhg.de

Case Study Objective

To demonstrate

the use of Building Block Development methodology

for development of

Billing Management building blocks

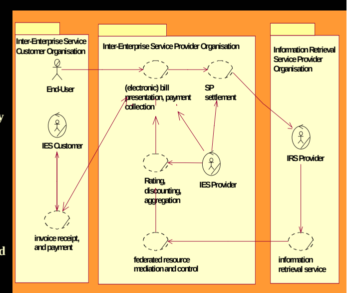
Case Study Outline

- **Requirement analysis: A brief introduction to Billing management process**
- **Definition of Billing System Architecture using Building Block (BB) Development Methodology Artefacts.**
- **Billing System in FORM Reference Architecture**
- **Billing System Boundary and Use Case Actors**
- **Billing Use Case Model**
- **Design Object Model: Building Block Diagram**
- **Example Contracts Interfaces**
- **Mapping Use Cases onto BB**
- **Boundary Information Model and Objects**
- **System Integration and Technology Architecture**
- **Technology Architecture Description**
- **Lessons Learnt, System Evaluation, Future Development**

Requirements Analysis

Key Requirements Addressed:

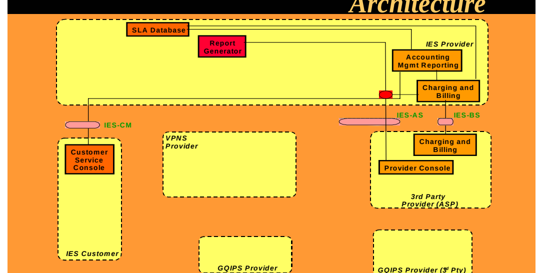
- **Support for convergence of services (voice and data)**
- **Real-time responses levels**
- **Adaptable service mediation**
- **Distribution of mediation facility**
- **Support for service value chain**
- **Charge aggregation (Composed Services)**
- **Automated inter SP domain accounting**
- **Interaction with legacy billing systems**
- **Demand for guaranteed QoS and related discounting**
- **Rapid service deployment**



Billing System Architecture (using Methodology Artefacts)

- Building Block (BB)
  - Federated Mediation Adaptor (FMA), Rating Bureau Service (RBS) et cetera
- Building Blocks Contracts
  - InterdomainAcctMgmt, RBSCtr, et cetera
  - interface signature encapsulated in XML
- BB and Contracts definitions:
  - Control, Boundary and Entity Classes
- Boundary Information Model
  - XML Schema Elements mapped onto UML Classes
- System processes integration
  - Primarily two differing sets of concerns: federated mediation and rating ... but contributing towards a common goal (billing)
  - Main operations: mediation, charging and settlement, and online billing
  - Use of TMForum for system process integration. IPDR (IP Detail Record) Organisation has also adopted the TMForum's Billing system processes for its own reference architecture.

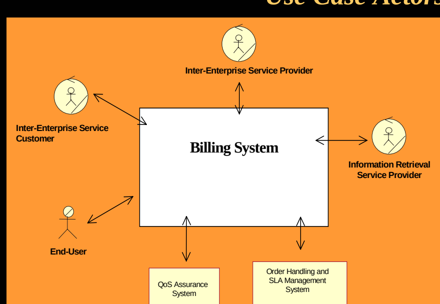
Billing System in FORM Reference Architecture



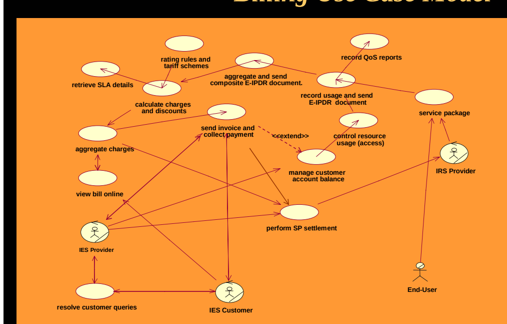
IES-BS: This reference point represents a business-to-business (B2B) relationship between the IES (Inter-Enterprise Service) Provider and one or more third-party SPs e.g., IRS (Information Retrieval Service) Provider.

IES-CM: This reference point represents a business-to-customer relationship between the IES Provider and the IES Customer.

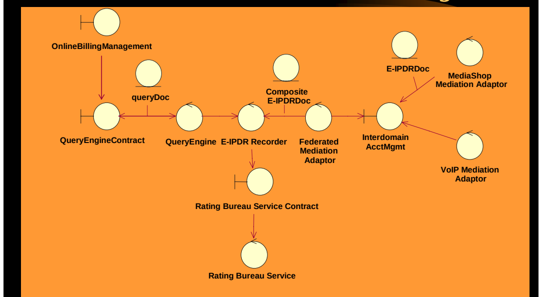
Billing System Boundary and Use Case Actors



Billing Use Case Model



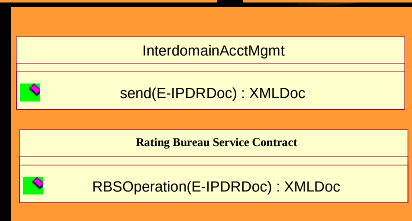
Design Object Model: Billing BB Diagram



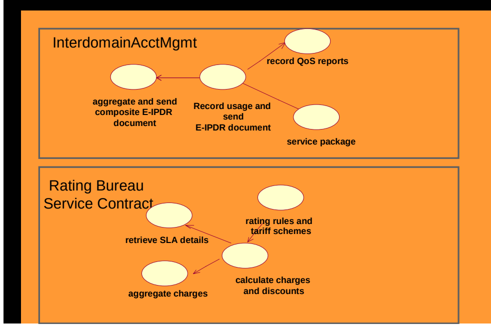
Example Contracts Interfaces

BB Contract Name: interdomainAcctMgmt
BB Name: Federated Mediation Adaptor
Brief Contract Description:
This contract allows….
Reference Points: IES-BS

BB Contract Name: **Rating Bureau Service Contract**
BB Name: Rating Bureau Service
Brief Contract Description:
This contract allows….
Reference Points: Internal



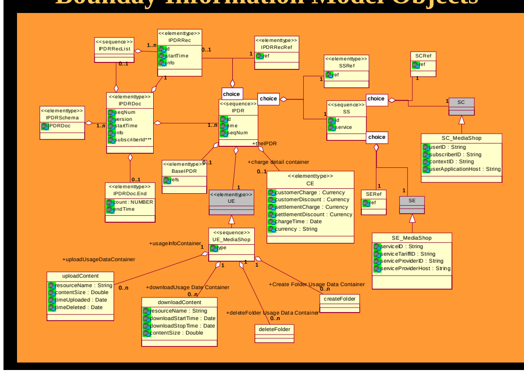
Example: Mapping Use Cases onto BB and Contracts



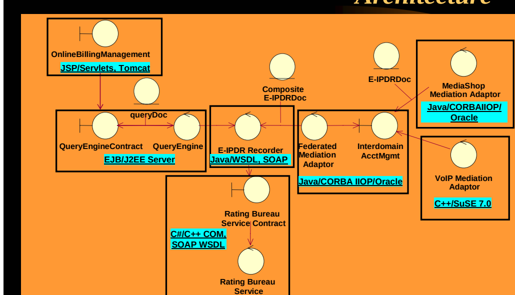
Billing Boundary Information Model

- Boundary Information Model is based on an *enhanced* Master IPDR (IP Detail Record) Schema. (IPDR specification version 2.6)
- IES-BS and IES-CM: Information exchanged at these reference point is in the form of an XML instance document, called called E-IPDR Document, of the enhanced Master IPDR Schema. (E-IPDR stands for *enhanced* IPDR.)
- E-IPDR Document is the information objects communicated at the interface of the Building Block Contracts (e.g., InterdomainAcctMgmt, IPDRecCtr)
- Information objects (mainly usage and charge details) are integrated under a single tree-shape structure. Integration is done at the level of Billing system boundary. Objects and structure are shared by all BB contracts, a *shared information schema* (see next slide).
- A particular "view" (or set of objects) of schema meets the requirements of a particular BB (or set of BBs) of Billing system.

Bounday Information Model Objects



System Integration and Technology Architecture



Technology Description

- Billing Building Blocks implemented as CORBA and EJB Components.
- Contracts definitions: Current mappings to specific technologies (CORBA IDL and WSDL)
- Interaction technologies: IIOP, RPC COM/CORBA-SOAP.
- All information accessed and passed through this contract is done in IPDR format.
- Technology Inter-operability (EJB with CORBA, XML and RDBMS)
- Boundary Information Model
  - Information objects specification: W3C XML Schema, Structure and Data Types
  - Information objects storage: ORACLE RDBMS

Lessons Learnt

- It is difficult to categorise BBs in three-tier architecture of moderately complex system.
- Contract supporting a Reference Point may possess more functionality then assigned during design phase.
  - Example: InterdomainAcctMgmt may also monitoring and recording network QoS deterioration
- Reference Point IES-BS may also include a contract that supports payment settlement (Please refer to use case model).
- Interfaces implementations are kept simple because of the more complex structures (defined in XML) that are passed.
- Standardised information model (IPDR Organisation) proved to be useful in developing Boundary Information Model.
  - Reconciles two differing sets of concerns (information-wise), service mediation and charge details that lead to a common goal, ie, Billing.
- A ***Contract Set*** and ***BB Group*** to perform value-based, QoS dependent charging and billing can be envisaged (A *Fulfilment-Assurance*-Billing Contract Set).

System Evaluation

- The Billing BB case study has been implemented in an EU funded R&D project - FORM. Tested in two trials. Results provided to standardisation bodies.
- Interworking between BB contracts.
- Functionality / Quality of BB
  - Test case result were mapped onto requirement captured in the requirement analysis phase.
- Applying of / Experiences with Framework, Methodology
  - Reference Architecture was slightly refined during evaluation process.
- Efficiency / Qualities of the overall System
  - XML Schemas more complicated than DTDs but are richer and useful in expressing a vocabulary of billing management business process.
- Boundary Info Model
  - needed to be enhanced to support service value chain involving several domains.
  - Mediation and charging of composite services need further study.

Future Developments

- Use of XML
  - XMI and XSLT in transforming technology neutral to technology specific transforms
  - XSLT to drive flexible model gateways
- Use of ebXML for federated accounting management business process
- Generic approach to aggregated services mediation:More control functionality to FMA BB.
- More investigation into Web service technologies (XSLT, WSDL, SOAP) for BB and contract definitions.
- Enhancements to Boundary Information Model (to include more different types of contracts)
- Enhancements to BB and contracts to support QoS and charge settlement
- Enhancements to BB and contracts to support a guaranteed delivery of IPDR documents.

QUESTIONS ……….

And Break !!!!

NOMS 2002

Federated Accouting Management System Demonstration

Thomas Gringel (*gringel@fokus.fhg.de*)

Fraunhofer FOKUS

FOKUS

SECTION III

Model-based Development of Management Systems for Management Business Process

Vincent P. Wade
Trinity College Dublin

GUIDELINE #2
Business Process Guidelines

Goal:

" …. To provide methodological guideline for construction of B2B and B2C management solutions that implement management business processes using Building Blocks."

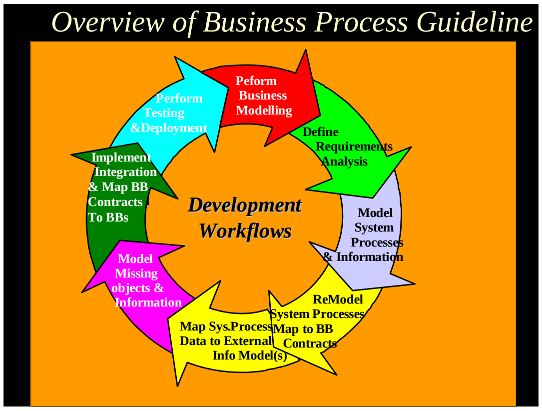
Intended users of Guideline:
- Management System Integrators.
- Service Providers.

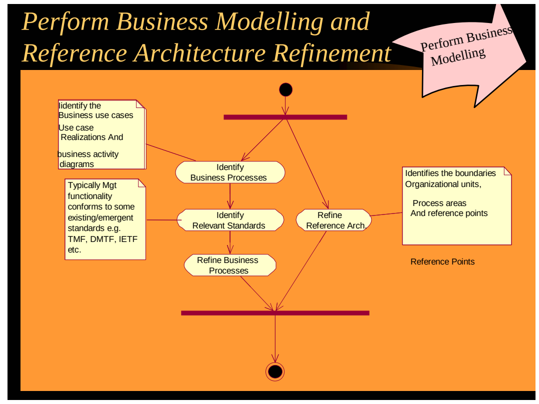
Objectives

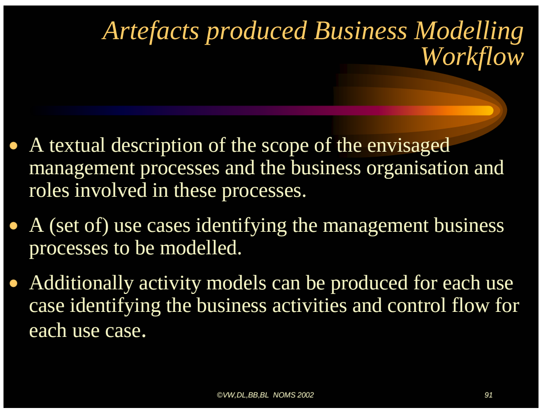
- To provide a development guideline which will allow telecom operators/service provider and management systems integrators to construct management solutions from Building Block Contracts.
- To provide support for a 'Business Process Driven' approach to management system construction from re-usable Building Blocks

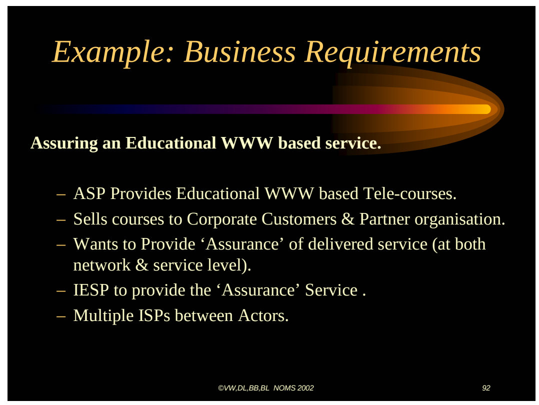
Some Assumptions

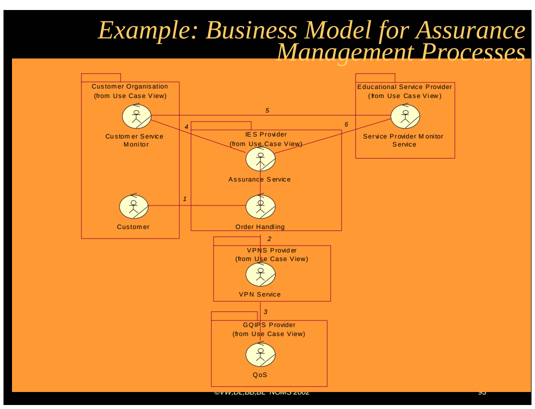
- A Catalogue of existing Building Block Contract Specifications and the Building Blocks which support them.
- Technology Gateways which provide protocol or technology mappings between Building Blocks developed using different technologies e.g. providing EJB Components interworking using IIOP and SOAP.

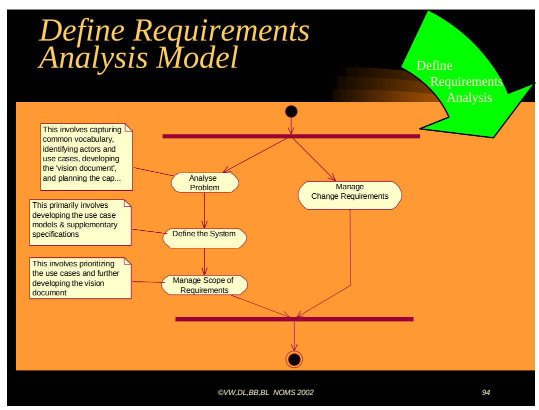

Overview of Business Process Guideline
Peform Business Modelling
Define Requirements Analysis
Model System Processes & Information
ReModel System Processes Map to BB Contracts
Map Sys.Process Data to External Info Model(s)
Model Missing objects & Information
Implement Integration & Map BB Contracts To BBs
Perform Testing &Deployment
Development Workflows

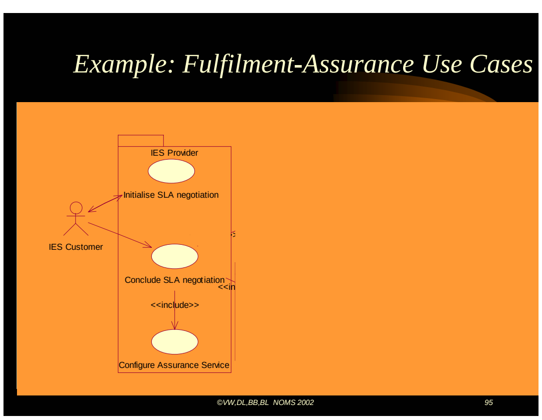

Perform Business Modelling and Reference Architecture Refinement
Perform Business Modelling
Identify the Business use cases Use case Realizations And business activity diagrams
Identify Business Processes
Typically Mgt functionality conforms to some existing/emergent standards e.g. TMF, DMTF, IETF etc.
Identify Relevant Standards
Refine Reference Arch.
Identifies the boundaries Organizational units, Process areas And reference points
Reference Points
Refine Business Processes

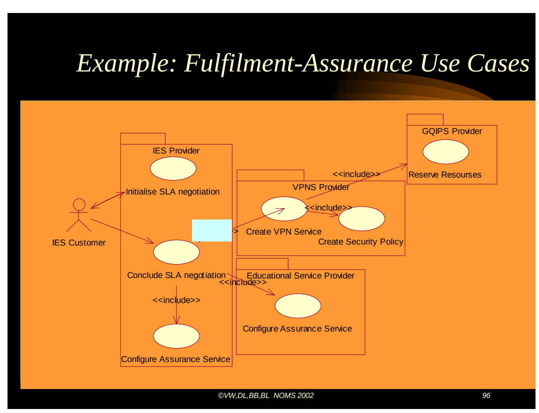

Artefacts produced Business Modelling Workflow
A textual description of the scope of the envisaged management processes and the business organisation and roles involved in these processes.
A (set of) use cases identifying the management business processes to be modelled.
Additionally activity models can be produced for each use case identifying the business activities and control flow for each use case.
©VW,DL,BB,BL NOMS 2002
91


Example: Business Requirements
Assuring an Educational WWW based service.
– ASP Provides Educational WWW based Tele-courses.
– Sells courses to Corporate Customers & Partner organisation.
– Wants to Provide 'Assurance' of delivered service (at both network & service level).
– IESP to provide the 'Assurance' Service .
– Multiple ISPs between Actors.
©VW,DL,BB,BL NOMS 2002
92


Example: Business Model for Assurance Management Processes
Customer Organisation (from Use Case View)
Customer Service Monitor
Customer
IES Provider (from Use Case View)
Assurance Service
Order Handling
Educational Service Provider (from Use Case View)
Service Provider Monitor Service
VPNS Provider (from Use Case View)
VPN Service
GQIPS Provider (from Use Case View)
QoS


Define Requirements Analysis Model
Define Requirements Analysis
This involves capturing common vocabulary, identifying actors and use cases, developing the 'vision document', and planning the cap...
Analyse Problem
Manage Change Requirements
This primarily involves developing the use case models & supplementary specifications
Define the System
This involves prioritizing the use cases and further developing the vision document
Manage Scope of Requirements
©VW,DL,BB,BL NOMS 2002
94


Example: Fulfilment-Assurance Use Cases
IES Provider
Initialise SLA negotiation
IES Customer
Conclude SLA negotiation
<<include>>
Configure Assurance Service
©VW,DL,BB,BL NOMS 2002
95


Example: Fulfilment-Assurance Use Cases
GQIPS Provider
Reserve Resourses
IES Provider
Initialise SLA negotiation
<<include>>
VPNS Provider
Create VPN Service
Create Security Policy
IES Customer
Conclude SLA negotiation
Educational Service Provider
Configure Assurance Service
©VW,DL,BB,BL NOMS 2002
96

Perform System Process & System Information Modelling

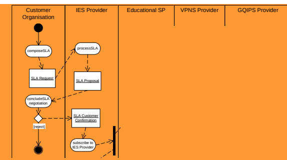
Model System Processes & Process Information

- Perform model of system activities, their control flows and their information flows, which occur in the management system processes to be developed.
- These management processes are identified from the use cases defined in the previous workflow.

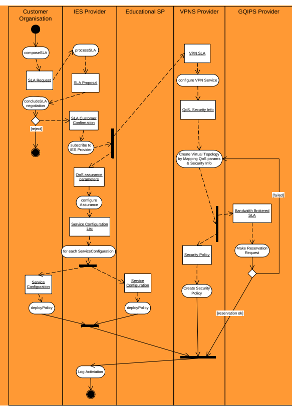
The models produced by the workflow are:

- Management System Process activities and Control Flows captured as Activity diagrams (graphs).
- Class Diagrams defining the information which flows between activities in the activity diagrams
- Collaboration diagram with analysis objects representing activities (defined in the activity diagrams) and object from the class diagrams representing data flows between these analysis objects.

Example:
Fulfilment
&Assurance
Activity Diagram



Example:
Fulfilment
&Assurance
Activity Diagram



Re-model System Processes; Map to Building Block Contracts workflow; Map System Process data to Boundary Information Model(s)

- Uses the activity definitions and diagrams to identify candidate Building Block Contracts, which offer equivalent/appropriate behaviours.
- Unlikely that exact matches of behaviour and the appropriate level of granularity will be readily available.

  => a re-organisation of the system development processes is necessary to decompose system activities to the granularity, which matches some of the Building Block Contract behaviours.
- Unlikely that all activities of a system process will be performed by Building Block Contracts.

  => missing functionalities need to be identified for later development effort in the guideline.

Remodel & Map Process Activities to BB Contracts

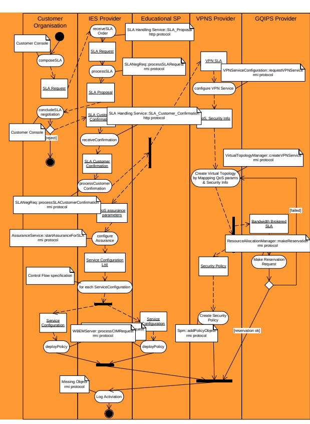
Re-model System Processes & Map to BB Contracts

- However, behavioural re-arrangement is not the only cause for reorganising the System Processes.
- Because of the likely mismatch between the Boundary Information Models of the Building Block Contracts and the information flows in the management processes, extra activities which perform data retrieval, transformation or generation will also be required.
- This remodelling work will need to be iterated a number of times until a satisfactory balance between reuse of Building Block Contracts and development of new software (i.e. missing functionality) is achieved.

Re-model System Processes & Map to BB Contracts

The result of this workflow is:

- A new set of system activity diagrams, in which each activity is noted as being supported by a (named) Building Block Contract or is noted as not supported.
- Activity and Collaboration Diagrams indicating information flows between the analysis objects each of which represents a Building Block Contracts in the catalogue, or a new object which needs to be implemented to support the management processes



Model Missing Objects and Information

- This workflow supports the modelling of the objects and information which are not supported by Building Block Contracts. The objects are identified in the collaboration diagram (representing the management business processes).
- The workflow follows standard software analysis and design activities for the modelling of the (classes and components) to support these objects

Implement Building Block Integration

- Facilitates the implementation of the necessary ‘application integration logic’.
- Implements the control and data flows represented in the system activity diagrams.
- The integration allows the invocation of Building Block Contract interfaces and the bespoke object interfaces (developed in the previous workflow).

©VW,DL,BB,BL NOMS 2002 105

Map Building Blocks Contracts to Building Blocks & deploy BBs and Business (Logic) Objects)

- Facilitates mapping of Building Block Contract onto Building Blocks.
- Building Blocks may be technology dependent and therefore the necessity of using technology or protocol gateways may be required to allow building blocks to interoperate
- Result is a full specification of the management system components as it cooperates to support the management processes

©VW,DL,BB,BL NOMS 2002 107

Map Building Blocks Contracts to Building Blocks & deploy BBs and Business (Logic) Objects)

- Activity diagrams where each activity is a component in the management system. The activity diagram will show the control flow of the system management process
- Class/object Model which models the information passed between the system components
- Collaboration Diagram with each component in the system modelled including the passing of information between the components

©VW,DL,BB,BL NOMS 2002 108

Perform Testing & Deployment

- Defines and executes the testing necessary for the management system.
- Involves the design and implementation of test plans as well as their evaluations.
- Involves planning and assembly of packages so the system can be deployed onto the target environment.
- Involves the planning and execution of the deployment of the Building Blocks and other object implementations to ensure the proper execution of the system

©VW,DL,BB,BL NOMS 2002 109

Case Study: Model-based Development of Component Based Management Systems for the Fulfilment Business Process

Birgitte Lønvig
Ericsson Denmark
Birgitte.Lonvig@ericsson.dk

Case Study

To demonstrate

the use of FORM Business Process Driven Development Methodology

for Development of

Fulfilment Management Systems

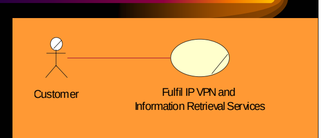
©VW,DL,BB,BL NOMS 2002 111

Tutorial Outline

- **Business Modelling**
  - **Business use case model**
  - **Business object model**
  - **Reference Model**
- **Requirements Analysis**
  - **Use case model**
- **System Process and Information Modelling**
  - **Activity diagram**
- **Mapping to Building Block Contracts**
  - **External Information model**
- **Modelling of Missing Objects**
- **BB Integration & mapping BB Contracts to BBs**
  - System Integration and Technology Architecture

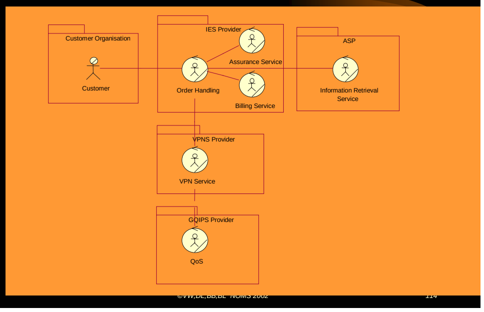
Business Use Case Model

Description:
- Order
- SLA negotiation
- Service Provisioning
- - Service Assurance
- - Service Billing

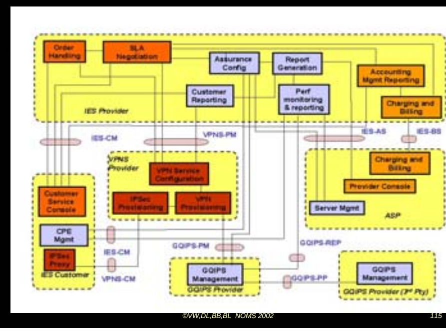


©VW,DL,BB,BL NOMS 2002 113

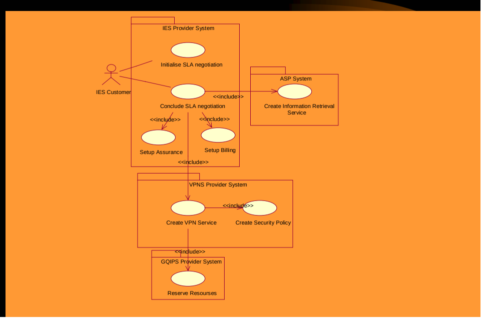
Business Object Model



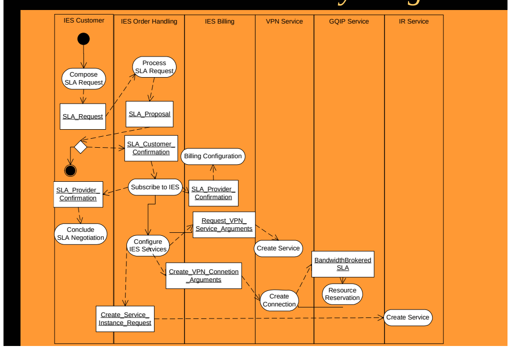
Reference Model



Use Case Model



Activity Diagram

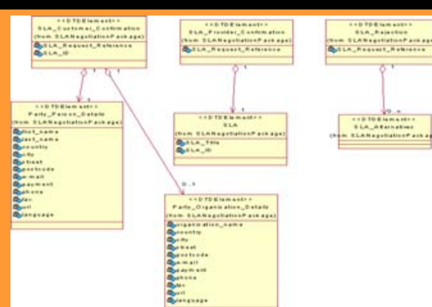


Mapping to BB Contracts

The customer negotiates and orders on-line via the IESP Order Handling Customer Interface a service package offered by the IESP consisting of a VPN service and an Information Retrieval service. Once the sale is agreed and confirmed the IESP Order Handling Process ensures that the subscription details are recorded and that the customer's service instances are provisioned

Contract Name	Description
SlaNegReq	This contract enables a party to enter and complete a SLA negotiation process with a SLA Negotiation Engine. The party entering the negotiation process is a prospective service customer. The SLA negotiation engine is able to control the negotiation process on behalf of a service provider.

External Information Model



Missing Objects ??

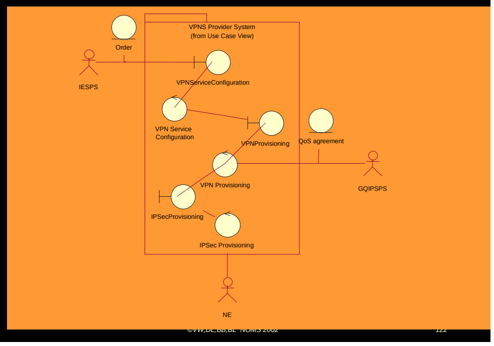
VPN Service – GQIPS

The external information model used to support the interactions between the VPNS system and GQIPS system in the fulfilment scenario corresponds to the GQIPS BB contract specification. No change of this contract has been necessary for adaptation to the sub-scenario.

Building Block Integration

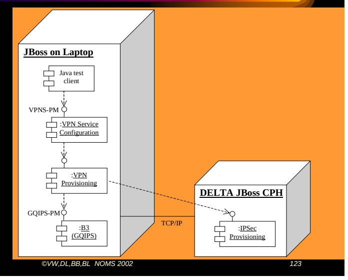
- Architectural styles used:
  - Layered style
  - Event Based style
  - Client-Server style
- Where to cover the control flow:
  - Control flow outside the BBs – use of a Work flow engine
  - Layered style:
    - One workflow engine for each layer
    - (Simpel case: Control flow inside BBs, if one BB per layer)

VPN BBs



Technology Architecture

- J2EE
  - (EJB, JMS, JNDI)
- XML



Evaluating the Fulfilment System

- The Fulfilment case study has been implemented in the EU funded R&D project - FORM. Tested in two trials. (*See demo in demo area*).
- Re-usability of BB and BB Contract
- Interworking between BB contracts
- Functionality / Quality of BB
  - Test case result were mapped onto requirement captured in the requirement analysis phase.
- Applying of / Experiences with ODF Framework, Methodology
- Efficiency / Qualities of the overall System

My Personal Views

Strong Points:
- Focus on Business Processes
- Reference model, reference points > contracts
- Clear separation of concerns:
  - Design for reuse = BB Development Guideline
  - Design by reuse = Business Process Driven System Development Guideline

Open Issues:
- Stronger focus on software architecture and non-functional requirements
- Stand-alone Methodology/Process?

Section IV

Lesson Learnt and Future Developments

Evaluating the Open Development Framework

- The Development Framework presented in section 2 builds on extensive management system development experience
- The Framework has recently been applied and evaluated in an EU funded R&D project – FORM
  - See http:/www.ist-form.org
- The two case studies and other examples were taken from the design documentation of this project
- We have captured some of the lesson learnt while developing and applying this Framework

Lessons Learnt: Architectural Balance

- A careful balance needed when defining a Development Framework:
  - Too prescriptive: excludes potential user with differing requirements
  - Not prescriptive enough: not much use to anyone, benefits of commonality lost
- Difficult to avoid technology biases:
  - Industry agreed architectures are often heavily influenced by those whose products a bound to a specific technology, e.g. CORBA, Java, MS

Lessons Learnt: Use of UML

- Useable methodology guidelines must take into account the restriction imposed by different CASE tools:
  - Support for all the aspects of the diagram types
  - Support for extension mechanisms, e.g. stereotypes
  - Limited model interchange between different tools
- Definition of development process
  - Must be limited: process must flexibly support organisation culture, expertise base and business aims
  - But can exploit common notations and similarities in approaches to define points were traced models can be produced in a common form to support exchange of wares

Future Developments: Modelling of Business Rules

- Modelling Business Rules:
  - Policies
  - Process flow
- Relationship of Rules to Building Block. Contracts and Systems
- Capturing Rules at different points in the development process
- Evolution of UML to support expression of Rules

©VW,DL,BB,BL NOMS 2002 130

Future Developments: Exploiting XML

- Use of XML for model publication and analysis on-line
  - Flexibly linking different models, e.g. contract interfaces and information content
  - Use of XSL in dynamically rendering different views of models
- Use of XMI and XSLT in transforming technology neutral model to technology specific ones

©VW,DL,BB,BL NOMS 2002 131

Conclusions

- Component reuse and legacy integration requires better understanding of the use and semantics of software interfaces
- Understanding needs to be shared between component developers, component reusers and interface standardisers
- Model driven development supports practical convergence of software development and exposure of models for shared understanding

©VW,DL,BB,BL NOMS 2002 132

Further Information

- FORM web site:
  - http://www.ist-form.org
- OSS/J
  - http://java.sun.com/products/oss/
- DMTF
  - http://www.dmtf.org
- NGOSS
  - http://www.tmforum.org
- UML
  - http://www.uml.org/
- OMG's Model Driven Architecture:
  - http://www.omg.org/mda/

Thank you …

and any questions ???

©VW,DL,BB,BL NOMS 2002 134

APPENDIX 3: Building Block Guideline Evaluation Form

Name of Participant (optional)

____

Indicate F-A-B Domain within which your BB was developed

____

Name(s) Building Block(s) which you were responsible/involved

____

Please complete the questions below. For each question there are five possible answers provided, illustrating a range of possibilities. Unless specifically indicated, please tick the box which most accurately reflects your choice within the range provided. If you have a comment/qualification on any of the questions please fill in the comment box provided.

1. Prior to the FORM project, please indicate your familiarity/expertise with Unified Modelling Language (i.e. UML Notations):

Expert *Never Used Before*

○ ○ ○ ○ ○

Optional Comment/Qualification:

2. **Prior to the FORM project, please indicate your familiarity/expertise with Rational Unified Process**

Expert *Never Used Before*

○ ○ ○ ○ ○

Optional Comment/Qualification:

3. **How responsible were you for the Building Block Development**

Was Principally Responsible *Review/Commented on Development*

○ ○ ○ ○ ○

Optional Comment/Qualification:

4. What role(s) did you principally play in the development of the BB ? (tick more than one role if appropriate)

Development	*Implementation*	*Managed*	*Reviewed Development*	*Other Developments*
○	○	○	○	○

Optional Comment/Qualification:

5. How useful were the Models/Artefacts (that you actually developed using UML/RUP) for describing your BB to other FORM partners/developers (leave box unticked if artefact not developed)

	Very Useful				*Useless*
BB Logical Architecture Diagram	○	○	○	○	○
	Very Useful				*Useless*
Reference Points Supported by BB	○	○	○	○	○
	Very Useful				*Useless*
Contract Supported by BB	○	○	○	○	○
	Very Useful				*Useless*
Contract description defining service provided by BB	○	○	○	○	○
	Very Useful				*Useless*
Use cases for BB	○	○	○	○	○
Collaboration diagram(s) to illustrate usage scope of BB					
	Very Useful				*Useless*

○ ○ ○ ○ ○

Very Useful *Useless*

Contract interface specification ○ ○ ○ ○ ○

Very Useful *Useless*

Information Object Model spec. ○ ○ ○ ○ ○

(UML Class Diagram of data objects exchanged at contract)

Very Useful *Useless*

Technological description for BB ○ ○ ○ ○ ○

Very Useful *Useless*

Collaboration diagrams for other BB interactions ○ ○ ○ ○ ○

where these BBs are relied upon by the BB being defined

Please comment on the use of BB contracts for describing BB to other developers/FORM partners:

Were there any extra artefacts which you would like to include in the BB description ?

6. How helpful was the general FORM Business Model when Developmenting your BB?

Highly Relevant *Not particularly Relevant*

○ ○ ○ ○ ○

Please Comment on your answer:

7 **How helpful was the development of the Use Cases for your Building Block?**

Very useful *Useless*

○ ○ ○ ○ ○

Optional Comment/Qualification:

8. **Identify the features of the Use cases that you used and rate their importance in your Development**

Very useful *Useless*

Use Case Diagram ○ ○ ○ ○ ○

Very useful *Useless*

Activity list ○ ○ ○ ○ ○

Very useful *Useless*

Supplementary Req'ments ○ ○ ○ ○ ○

Please comment on any aspect of the Use cases that caused difficulty:

9. How useful were the G909 requirements specification template for your BB Developments

Very relevant *Totally irrelevant*

○ ○ ○ ○ ○

Optional Comment/Qualification:

10. How useful were the 3 tiers (HIT,PAT, EIT) in the early analysis and identification of your building block(s):

Very useful *Very poor*

○ ○ ○ ○ ○

Optional Comment/Qualification:

11. How useful were the 3 tiers in the detailed Development of your building block(s):

Very clearly *Very unclearly*

○ ○ ○ ○ ○

Optional Comment/Qualification:

12. **Identify the usefulness of the following in determining the granularity/boundary of your building blocks**

Very useful *Useless*

HIT/PAT/EIT layers ○ ○ ○ ○ ○

Degree of self-contained behaviour (logical grouping of closely related behaviours)

Very useful *Useless*

○ ○ ○ ○ ○

Level of inter-dependence *Very useful* *Useless*

between grouping of classes ○○ ○ ○ ○

and interacting classes
(based on original use cases)

Identification of a 'service' *Very useful* *Useless*

or 'services' the BB could ○ ○ ○ ○ ○

uniquely support

Potential as Unit of *Very useful* *Useless*

Deployment ○ ○ ○ ○ ○

Potential as Unit of *Very useful* *Useless*

Manageability of ○ ○ ○ ○ ○

of the BB.

Please comment on any other criteria/aspect of the analysis/Development model you used to identify your BB boundary/granularity:

13. Identify the usefulness of the following MODELS in DEVELOPING your BB

	Very high				*Very low*
Use Case Realisation	○	○	○	○	○
	Very high				*Very low*
Object Model	○	○	○	○	○
	Very high				*Very low*
Collaboration Model	○	○	○	○	○
	Very high				*Very low*
Interface Signature	○	○	○	○	○

Please comment on any other criteria/aspect of the model you used in Developing your BB:

14. Did you have any difficulty in tracing the information flows between the computational objects <u>in</u> your BB

Very Difficult *No Difficulty*

○ ○ ○ ○ ○

<u>Please comment</u>:

15. Did you have any difficulty in co-ordinating/tracing the Shared Data which your BB needed for execution? (by shared data we mean information objects which need to exist/be known about externally from the BB)

Very Difficult *No Difficulty*

○ ○ ○ ○ ○

<u>Please comment</u>:

16. How useful did you find the concept of 'BUILDING BLOCK CONTRACT' in Developmenting your building block

High level *Very low*

○ ○ ○ ○ ○

<u>Please comment</u> on your experience in defining the Building Block Contract:

17. How difficult did you find writing the Building Block Description (in annexs of D5)

Very Difficult *Very Easy*

○ ○ ○ ○ ○

<u>Please comment</u> on any particular difficulty you experienced in Developmenting your BB:

18. Having developed a Building Block for Trial 1, what features of the development process require more explanation/guidance (indicate within each box provided what issues were not adequately dealt with).

Business Modelling

Contract Description/Type

Use Cases Descriptions

Object Model (for information passed externally to/from the BB)

Technology Description (for BB Implementation)

Testing of BB functionality

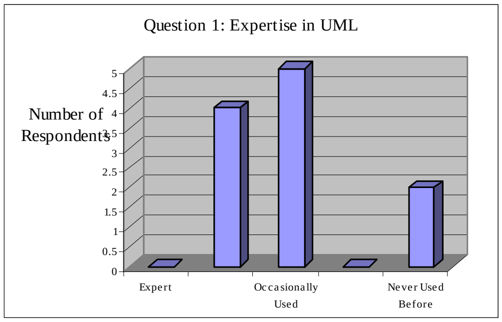
19. Were there any aspects of the Building Blocks Development which you think were not covered in the BB development process?

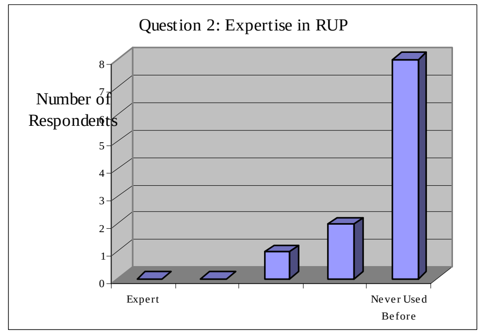
Please give details:

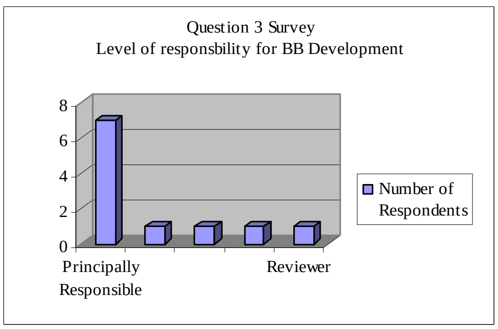
20. **Any other comments:**

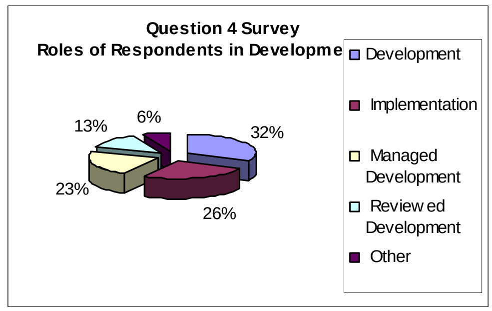
__Thank you for completing this questionnaire. The information will be used to further develop the FORM Development Guidelines__

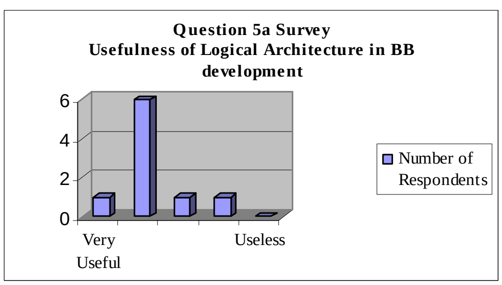
APPENDIX 4: Graphs for Building Block Development Guideline Evaluation Results

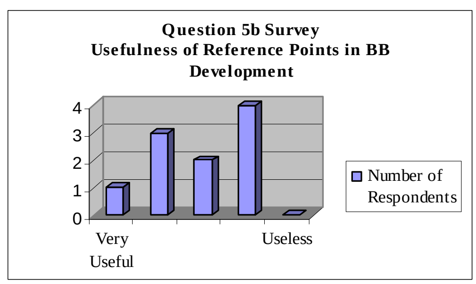

Question 1: Expertise in UML
Number of Respondents
Expert
Occasionally Used
Never Used Before

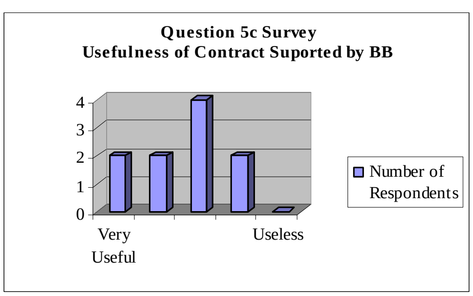

Question 2: Expertise in RUP
Number of Respondents
Expert
Never Used Before

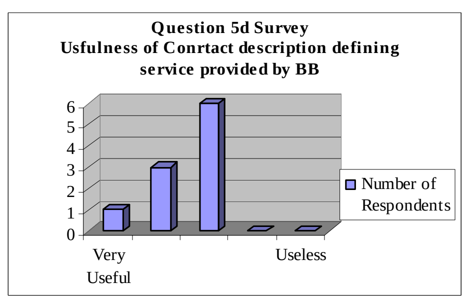

Question 3 Survey
Level of responsbility for BB Development
Principally Responsible
Reviewer
Number of Respondents

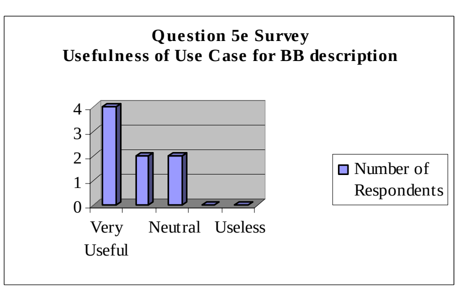

Question 4 Survey
Roles of Respondents in Developme
32%
26%
23%
13%
6%
Development
Implementation
Managed Development
Review ed Development
Other

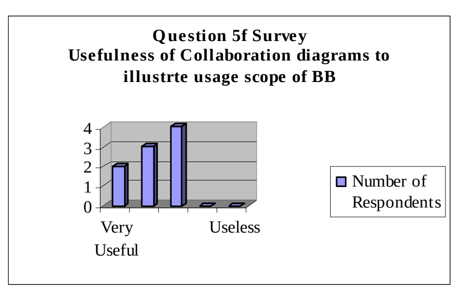

Question 5a Survey
Usefulness of Logical Architecture in BB development
Very Useful
Useless
Number of Respondents

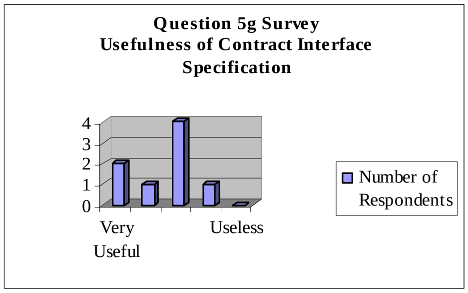

Question 5b Survey
Usefulness of Reference Points in BB Development
Very Useful
Useless
Number of Respondents

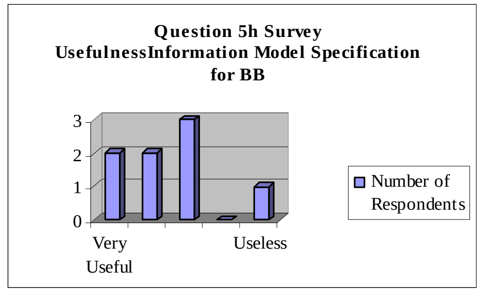

Question 5c Survey
Usefulness of Contract Suported by BB
Very Useful
Useless
Number of Respondents

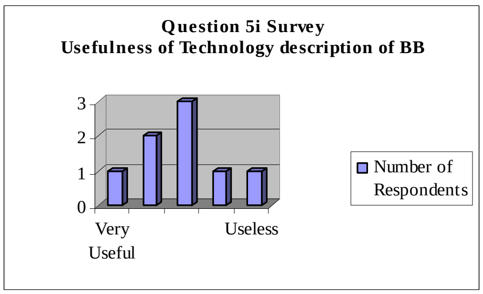

Question 5d Survey
Usfulness of Conrtact description defining service provided by BB
Very Useful
Useless
Number of Respondents

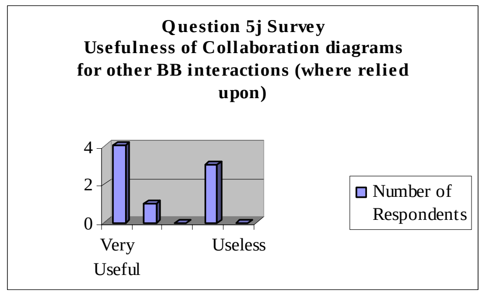

Question 5e Survey
Usefulness of Use Case for BB description
Very Useful
Neutral
Useless
Number of Respondents

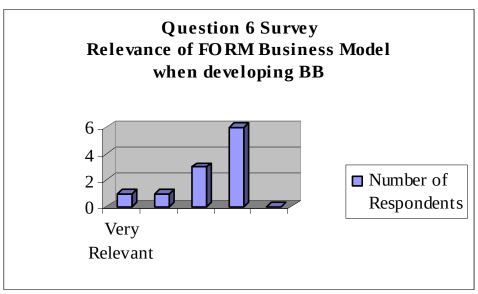

Question 5f Survey
Usefulness of Collaboration diagrams to illustrte usage scope of BB
Very Useful
Useless
Number of Respondents

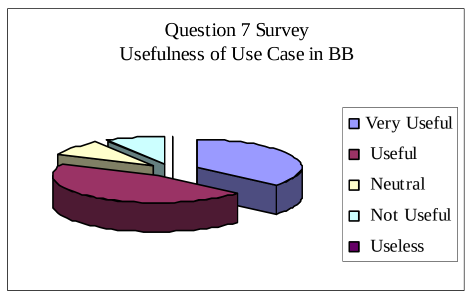

Question 5g Survey
Usefulness of Contract Interface Specification
4
3
2
1
0
Very Useful
Useless
Number of Respondents

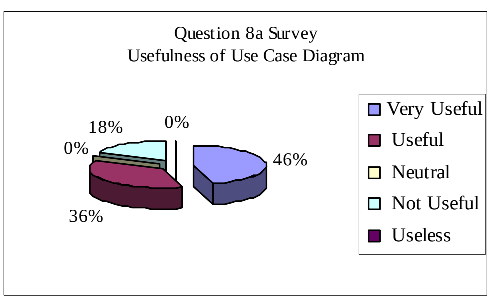

Question 5h Survey
UsefulnessInformation Model Specification for BB
3
2
1
0
Very Useful
Useless
Number of Respondents

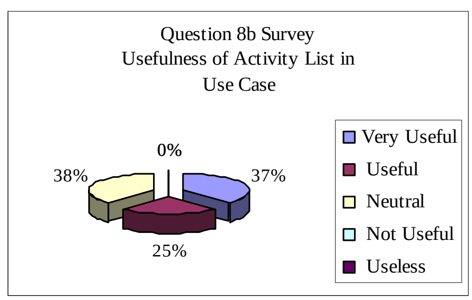

Question 5i Survey
Usefulness of Technology description of BB
3
2
1
0
Very Useful
Useless
Number of Respondents

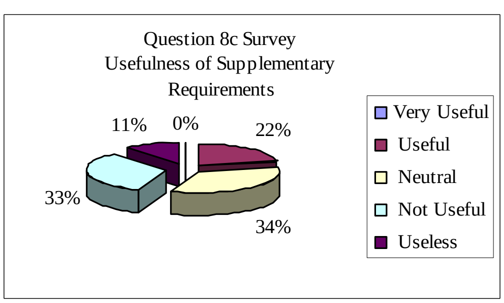

Question 5j Survey
Usefulness of Collaboration diagrams for other BB interactions (where relied upon)
4
2
0
Very Useful
Useless
Number of Respondents

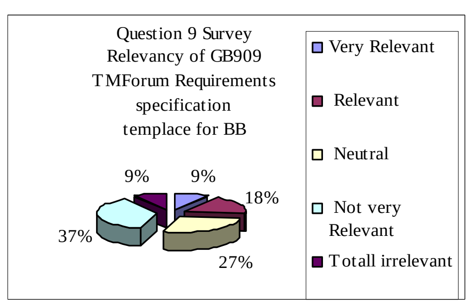

Question 6 Survey
Relevance of FORM Business Model when developing BB
6
4
2
0
Very Relevant
Number of Respondents

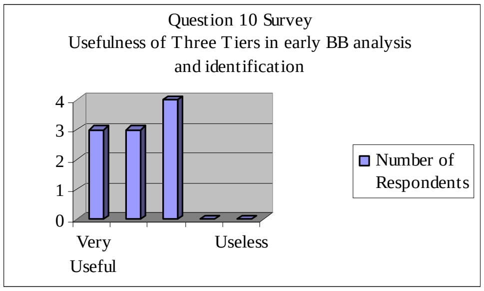

Question 7 Survey
Usefulness of Use Case in BB
Very Useful
Useful
Neutral
Not Useful
Useless

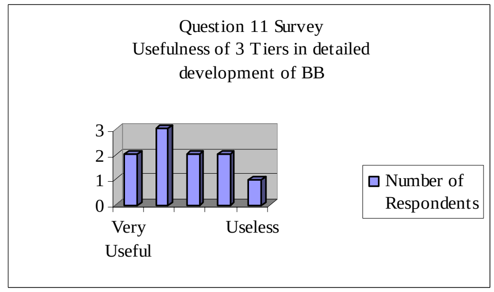

Question 8a Survey
Usefulness of Use Case Diagram
18%
0%
0%
46%
36%
Very Useful
Useful
Neutral
Not Useful
Useless

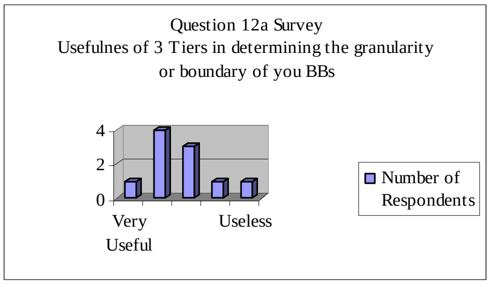

Question 8b Survey
Usefulness of Activity List in Use Case
0%
38%
37%
25%
Very Useful
Useful
Neutral
Not Useful
Useless

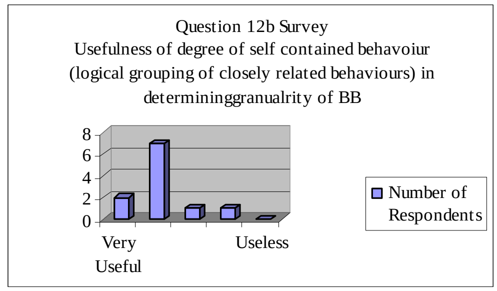

Question 8c Survey
Usefulness of Supplementary Requirements
11%
0%
22%
33%
34%
Very Useful
Useful
Neutral
Not Useful
Useless

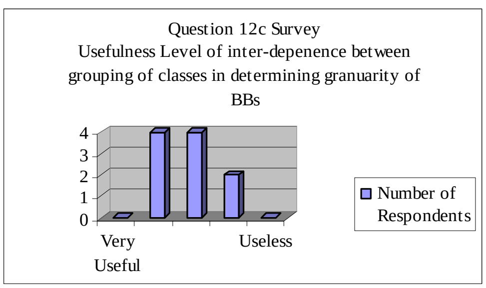

Question 9 Survey
Relevancy of GB909 TMForum Requirements specification templace for BB
9%
9%
18%
37%
27%
Very Relevant
Relevant
Neutral
Not very Relevant
Totall irrelevant

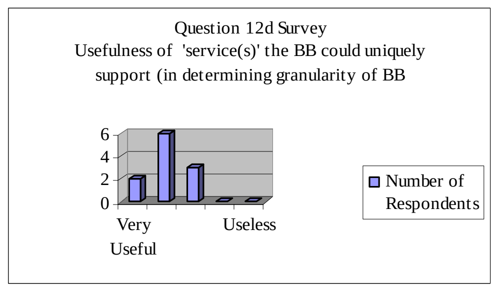

Question 10 Survey
Usefulness of Three Tiers in early BB analysis and identification
4
3
2
1
0
Very Useful
Useless
Number of Respondents

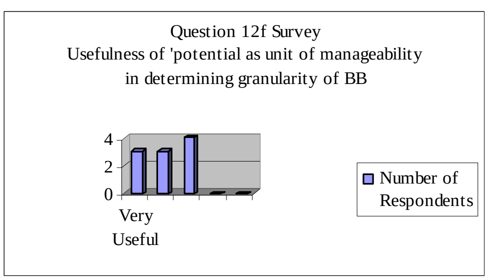

Question 11 Survey
Usefulness of 3 Tiers in detailed development of BB
3
2
1
0
Very Useful
Useless
Number of Respondents

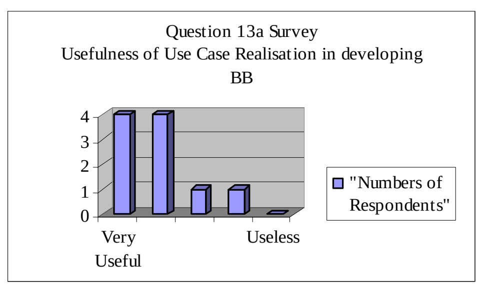

Question 12a Survey
Usefulnes of 3 Tiers in determining the granularity or boundary of you BBs
4
2
0
Very Useful
Useless
Number of Respondents

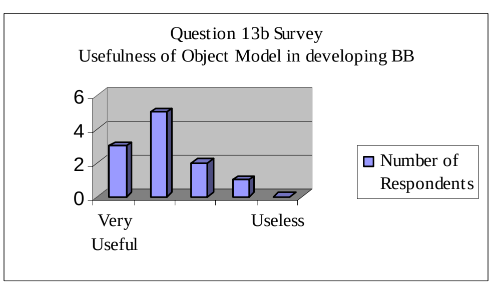

Question 12b Survey
Usefulness of degree of self contained behavoiur (logical grouping of closely related behaviours) in determininggranualrity of BB
8
6
4
2
0
Very Useful
Useless
Number of Respondents

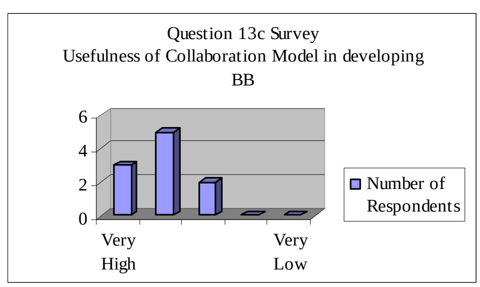

Question 12c Survey
Usefulness Level of inter-depenence between grouping of classes in determining granuarity of BBs
4
3
2
1
0
Very Useful
Useless
Number of Respondents

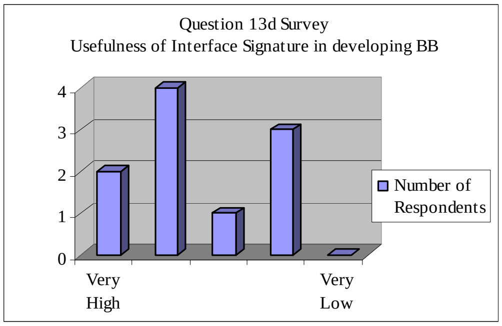

Question 12d Survey
Usefulness of 'service(s)' the BB could uniquely support (in determining granularity of BB
6
4
2
0
Very Useful
Useless
Number of Respondents

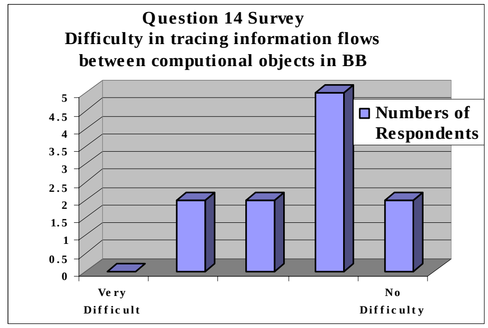

Question 12f Survey
Usefulness of 'potential as unit of manageability in determining granularity of BB
4
2
0
Very Useful
Number of Respondents

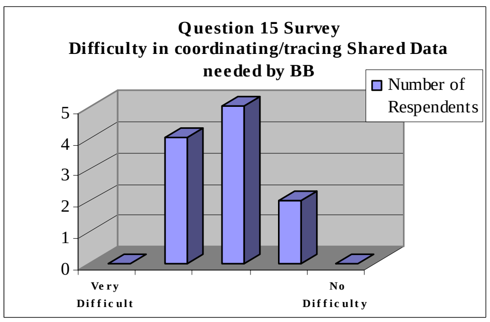

Question 13a Survey
Usefulness of Use Case Realisation in developing BB
4
3
2
1
0
Very Useful
Useless
"Numbers of Respondents"

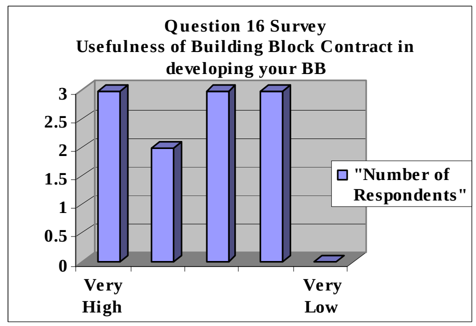

Question 13b Survey
Usefulness of Object Model in developing BB
6
4
2
0
Very Useful
Useless
Number of Respondents

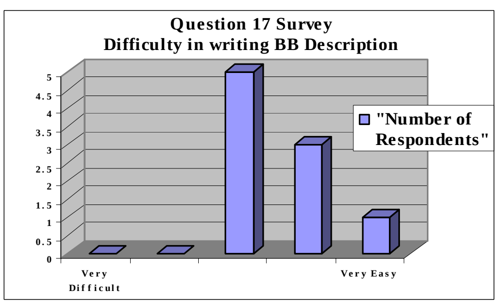

Question 13c Survey
Usefulness of Collaboration Model in developing BB
6
4
2
0
Very High
Very Low
Number of Respondents


Question 13d Survey
Usefulness of Interface Signature in developing BB
0
1
2
3
4
Very High
Very Low
Number of Respondents


Question 14 Survey
Difficulty in tracing information flows between computional objects in BB
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Very Difficult
No Difficulty
Numbers of Respondents


Question 15 Survey
Difficulty in coordinating/tracing Shared Data needed by BB
0
1
2
3
4
5
Very Difficult
No Difficulty
Number of Respendents


Question 16 Survey
Usefulness of Building Block Contract in developing your BB
0
0.5
1
1.5
2
2.5
3
Very High
Very Low
"Number of Respondents"


Question 17 Survey
Difficulty in writing BB Description
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Very Difficult
Very Easy
"Number of Respondents"

APPENDIX 5: Business Process Driven System Modelling Guideline Evaluation Form

Name of Participant (optional): ______________________________________

Indicate Systems for which modelling work was performed:e.g. F – A – B and/or modelling work - FA – FB – AB:

____

Please indicate the Management System(s) with which you were involved (Trial and/or MCG): ______________________________________

Please complete the questions below. For each question there are four possible answers provided, illustrating a range of possibilities. Unless specifically indicated, please tick the box which most accurately reflects your choice within the range provided. If you have a comment/qualification on any of the questions please fill in the comment box provided. The questions relate to the modelling work performed for the TRIAL systems and/or the MCG work.

1. **Prior to the FORM project, please indicate your familiarity/expertise with Unified Modelling Language (i.e. UML Notations):**

Expert *Frequent use* *Occasional Use* *Never Used Before*

○ ○ ○ ○

Optional Comment/Qualification:

2. **Prior to the FORM project, please indicate your familiarity/expertise with Rational Unified Process**

Expert *Frequent use* *Occasional Use* *Never Used Before*

○ ○ ○ ○

Optional Comment/Qualification:

3. **How responsible were you for the System Modelling/Development?**

Was Principally Responsible *Responsible for subsystem or sub process development* *Contributed to overall design/development* *Review/Commented on Development*

○ ○ ○ ○

Optional Comment/Qualification:

4. **What role(s) did you principally play in the modelling/development of the Business Process Driven System ? (tick more than one role if appropriate)**

Business Modelling/ Requirement Sepc.	*Design*	*Implementation*	*Managed /Reviewed Development*
○	○	○	○

Optional Comment/Qualification:

5. **How helpful was the general FORM Business Model when Developing your system?**

Highly Relevant	*Useful*	*Marginally useful*	*Not Relevant*
○	○	○	○

Please Comment why/why its was/wsa not useful:

6. **When performing the Business Modelling Workflow, please rate the usefulness of the following: (Note: this work may have been done by some partners during an earlier phase of the Project i.e. during D4 or during MCG or trial development work)**

	Highly Relevant	*Useful*	*Marginally useful*	*Not Relevant*
FORM Reference Architecture	○	○	○	○

	Highly Relevant	*Useful*	*Marginally useful*	*Not Relevant*
FORM Business Use Case(s)	○	○	○	○

	Highly Relevant	*Useful*	*Marginally useful*	*Not Relevant*
FORM Business Activity Diagrams	○	○	○	○

Please Comment why/why they were/were not useful:

7 How helpful was the development of the Use Cases for specifying your requirements for the MCG work or trial System development?

Very useful *Useless*

○ ○ ○ ○

Optional Comment/Qualification:

9. **Identify the notation that you used (for requirements capture) and rate their importance in your Development**

	Very useful			*Useless*
Use Case Diagram	○	○	○	○
	Very useful			*Useless*
Activity diagram & specification	○	○	○	○
	Very useful			*Useless*
Supplementary Req'ments	○	○	○	○

Please comment on any aspect of the Use cases that caused difficulty:

9. How useful were the G909 requirements for your Modelling/Development

Very relevant			*Totally irrelevant*
○	○	○	○

Optional Comment/Qualification:

10. **How useful was the separation of the activities into 3 tiers (HIT,PAT, EIT) in the <u>early analysis and identification</u> of your system modelling ?**

Very useful *Very poor*

○ ○ ○ ○

Optional Comment/Qualification:

11. When modelling the System Process, how useful were the UML activity diagrams in describing the control for for the intended system?

Very good in capturing system process control flow — *Very poor in capturing system process control flow*

○ ○ ○ ○

Optional Comment/Qualification:

12. How useful were the (system process) activity diagrams in describing the <u>data flow</u> for the intended system?

Very good in capturing system process data flow — *Very poor in capturing system process data flow*

○ ○ ○ ○

Optional Comment/Qualification:

13. How useful was the External Information Model in representing the shared information for the intended system?

Very good *Very poor*

○ ○ ○ ○

Optional Comment/Qualification:

14. How difficult was it to map system activities onto pre-defined Building Block Contracts?

Very Little Difficulty *Very difficult*

○ ○ ○ ○

Optional Comment/Qualification:

15. How difficult was it to map system information flows onto pre-defined Building Block Contracts Boundary Information Models?

Very Little Difficulty *Very difficult*

○ ○ ○ ○

Optional Comment/Qualification:

16. How difficult was it to design the Collaboration Diagram(s) which represented the BB Contact intractions (based on the system activity diagrams).

Very Little Difficulty *Very difficult*

○ ○ ○ ○

Optional Comment/Qualification:

17. Did the BB Contract Collaboration Diagram(s) represent/describe enough information regarding the interactions between BB Contracts which support the system process to be implemented?

Perfect level of Represention *Very Poor level of represention*

○ ○ ○ ○

Optional Comment/Qualification:

18. Did you have to model one or more system objects to provide extra functionality which was not supported by existing Building Block Contracts ?

YES ○ **NO** ○

If YES then comment on the difficulty of integrating this modelling work into the BB Contract collaboration diagram (representing the BB Contract interactions).

Very Little Difficulty *Very difficult*

○ ○ ○ ○

Optional Comment/Qualification:

19. How difficult was it to map the Building Block Contracts Collaboration diagrams onto a BB Collaboration diagram (ready for implementation)

Very Little Difficulty *Very difficult*

○ ○ ○ ○

Optional Comment/Qualification:

20. How long did you spend studying the Business Process Driven Development Guideline?

No Time At all	*Just learned it from the project presentations*	*Spent between . 1-5 hrs Reading Guideline*	*Spent Greater than 5 reading guideline*
○	○	○	○

Optional Comment/Qualification:

21 How useful were the Models/Artefacts (that you actually developed), for designing and modelling your management system (leave box unticked if artefact not developed)

Very useful *Not Useful at all*

Business

Use case modelling ○ ○ ○ ○

Business Activity *Very useful* *Not Useful at all*

(process) Modelling	o	o	o	o
System activity	*Very useful*			*Not Useful at all*
(process) models	o	o	o	o
BB Contract	*Very useful*			*Not Useful at all*
Shared Information	o	o	o	o
Models				
BB Contract	*Very useful*			*Not Useful at all*
Collaboration Diagrams	o	o	o	o
BB Collaboration	*Very useful*			*Not Useful at all*
Diagrams	o	o	o	o

Please comment on the usefulness of artefacts for describing the system to other developers/FORM partners:

Were there any extra artefacts which you would like to include in the BB description ?

22. Was the methodology clear in describing the following workflows in the system development cycle

	Very Clear			*Very Unclear*
Business Use case modelling	○	○	○	○
Business Activity (process) Modelling	○	○	○	○
System activity (process) models	○	○	○	○
BB Contract Collaboration Diagrams	○	○	○	○
BB Contract Shared Information Models	○	○	○	○
BB Collaboration Diagrams	○	○	○	○

23. Having Modelled or developed System (e.g. for Trial 2 or the MCG) what features of the development process require more explanation/guidance (indicate within each box provided what issues were not adequately dealt with).

Business Modelling

Requirements Analysis

System Process Modelling

Shared Information Modelling

BB Contract Colaboration Modelling

BB Collaboration Modelling

Testing of BB functionality

24. **Were there any aspects of the System Modelling/ Development which you think were not covered in the Guideline that should have been covered?**

Please give details:

____

____

____

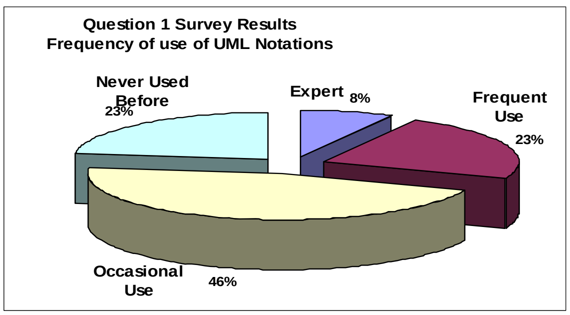
25. **Any other comments:**

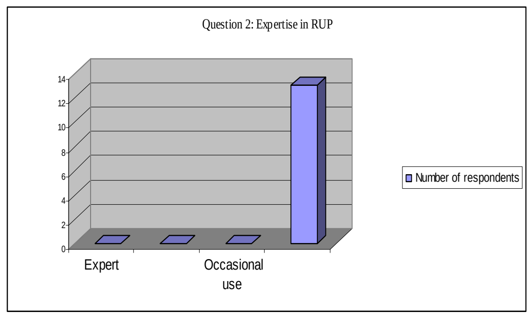
____

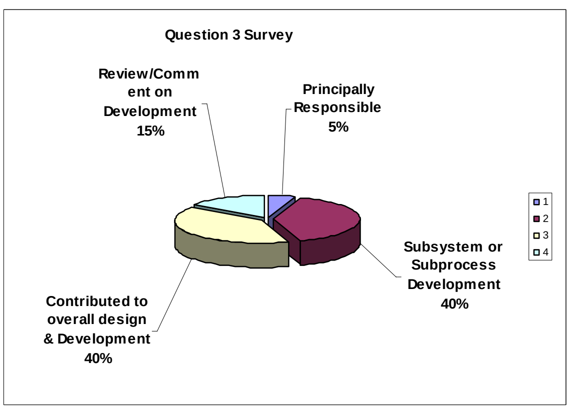
____

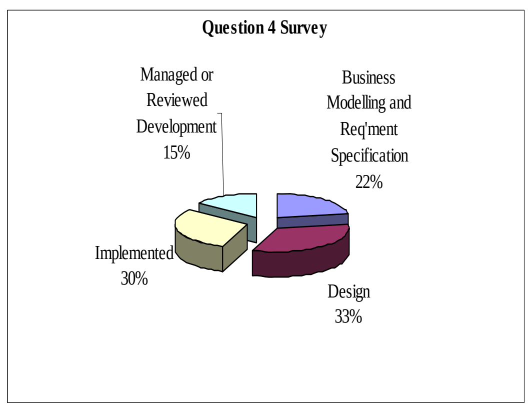
Thank you for completing this questionnaire.

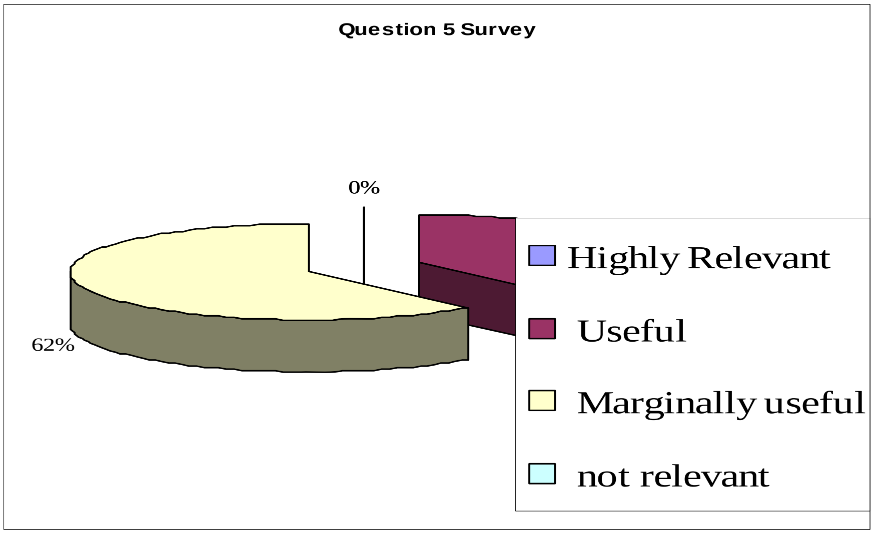
APPENDIX 6: Graphs for Business Process Driven Development Guideline Evaluation Results

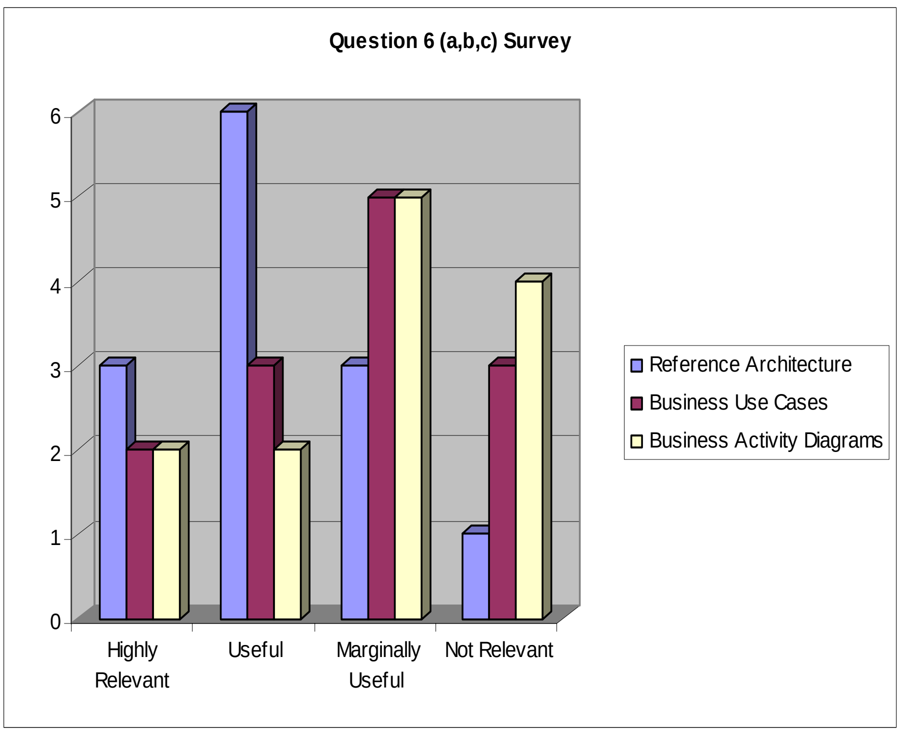


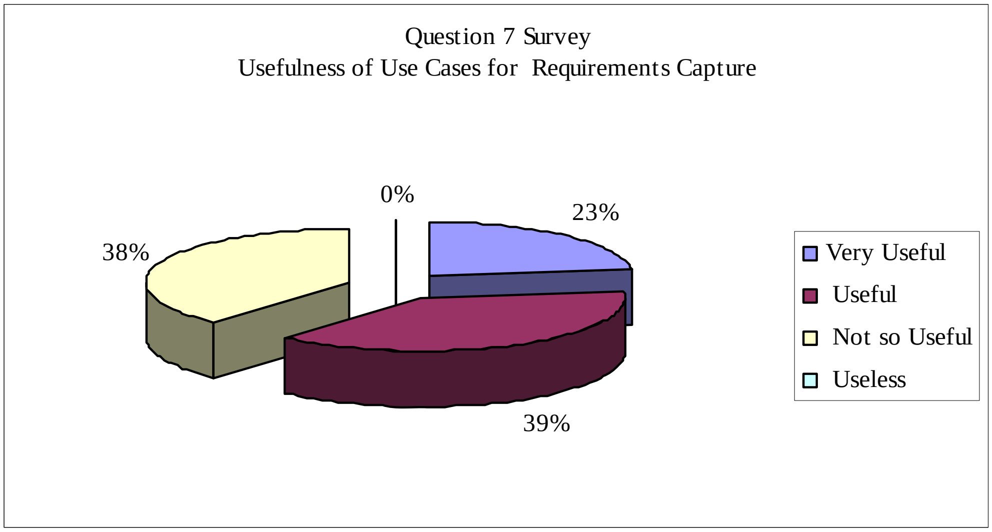


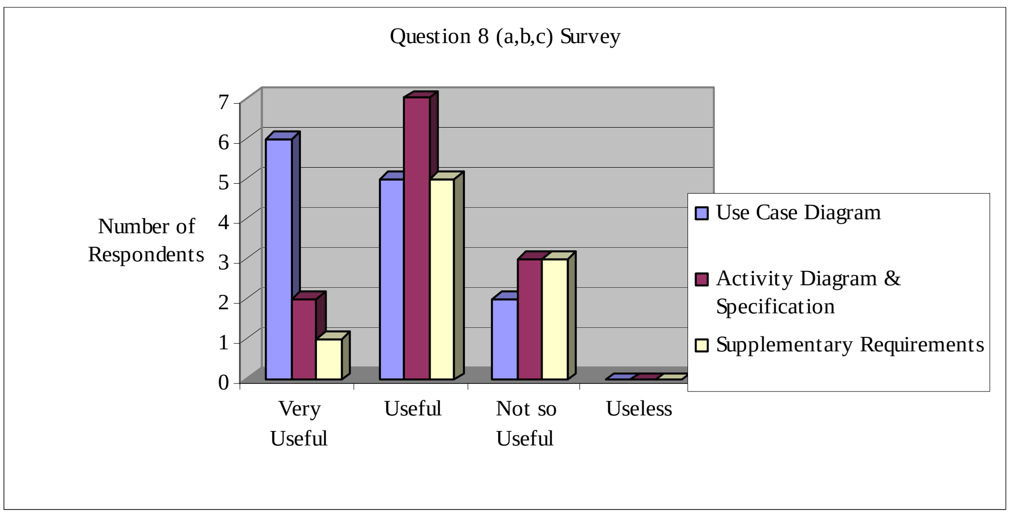


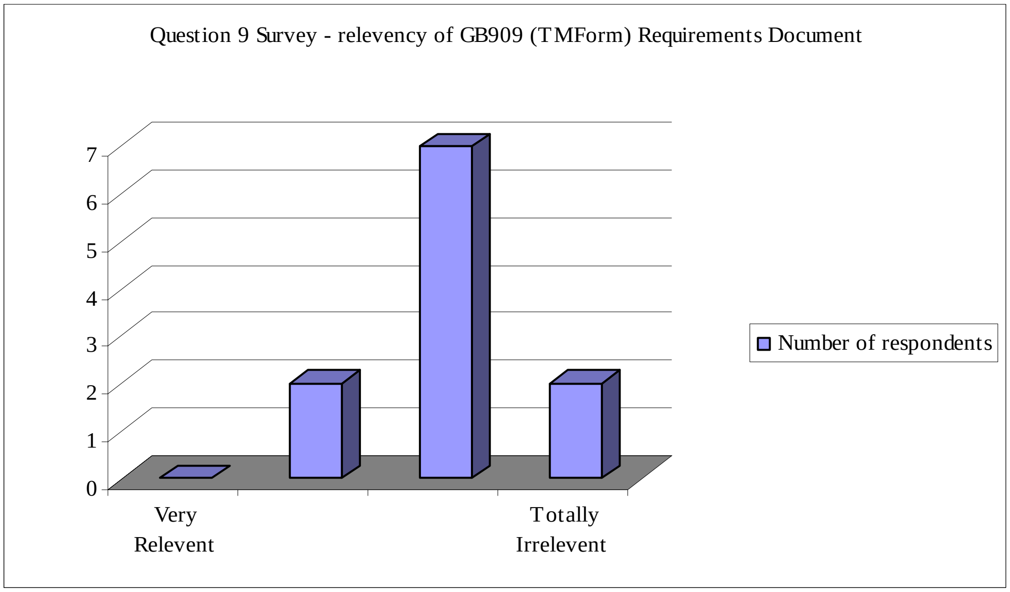


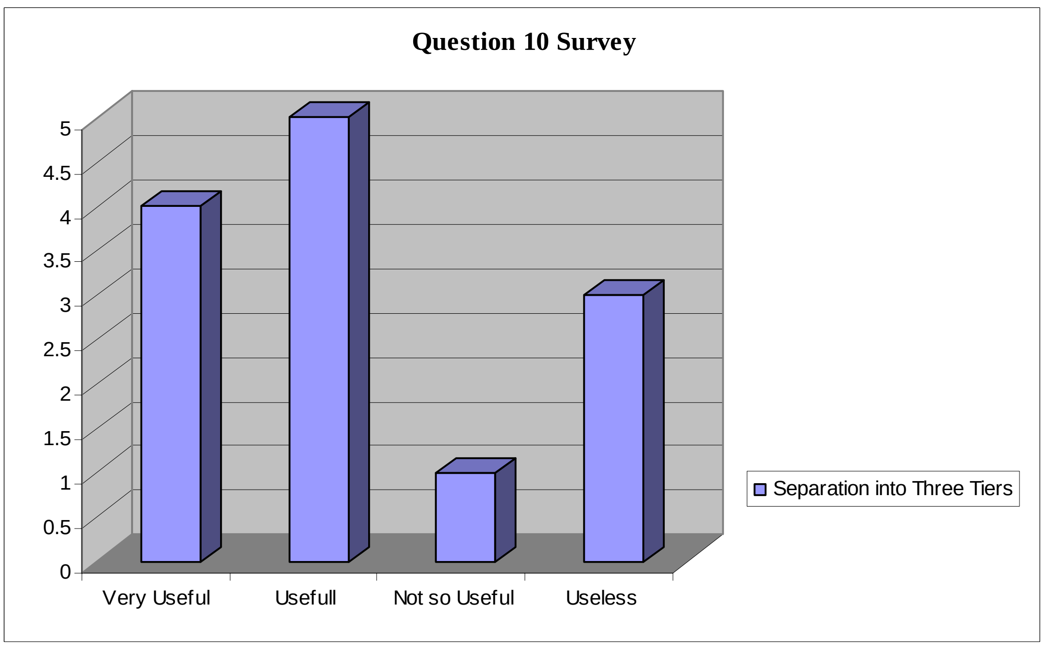

Question 5 Survey
0%
62%
Highly Relevant
Useful
Marginally useful
not relevant

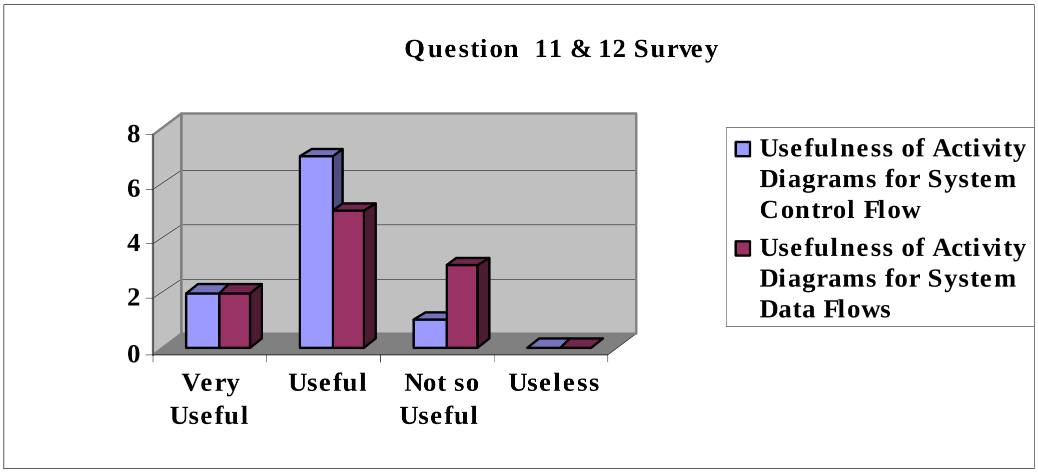

Question 6 (a,b,c) Survey
6
5
4
3
2
1
0
Highly Relevant
Useful
Marginally Useful
Not Relevant
Reference Architecture
Business Use Cases
Business Activity Diagrams

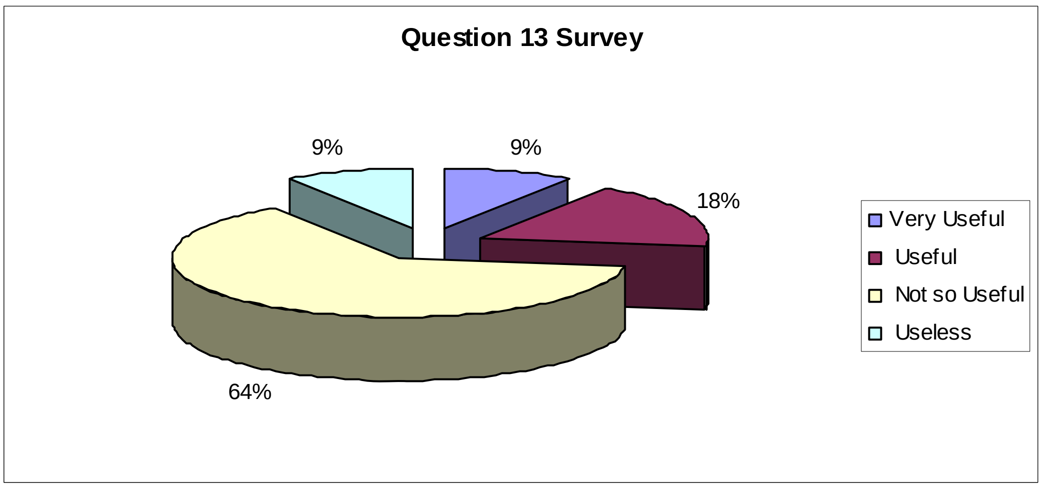

Question 7 Survey
Usefulness of Use Cases for Requirements Capture
0%
23%
38%
39%
Very Useful
Useful
Not so Useful
Useless

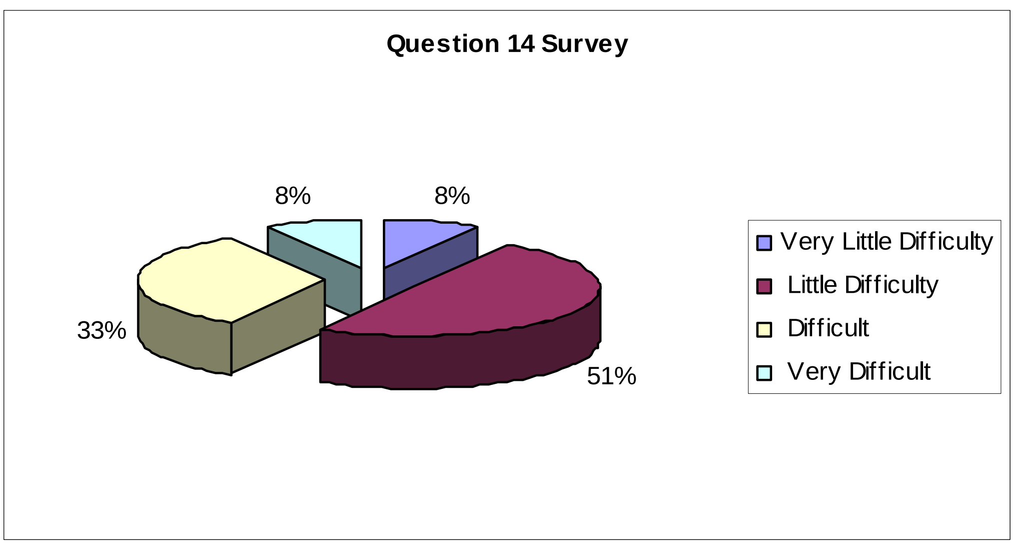

Question 8 (a,b,c) Survey
Number of Respondents
7
6
5
4
3
2
1
0
Very Useful
Useful
Not so Useful
Useless
Use Case Diagram
Activity Diagram & Specification
Supplementary Requirements

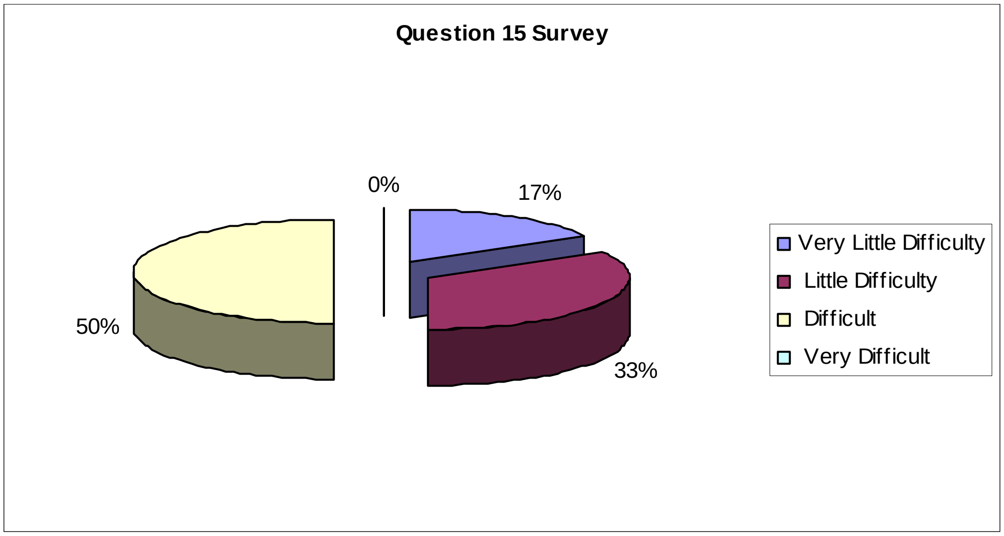

Question 9 Survey - relevency of GB909 (TMForm) Requirements Document
7
6
5
4
3
2
1
0
Very Relevent
Totally Irrelevent
Number of respondents

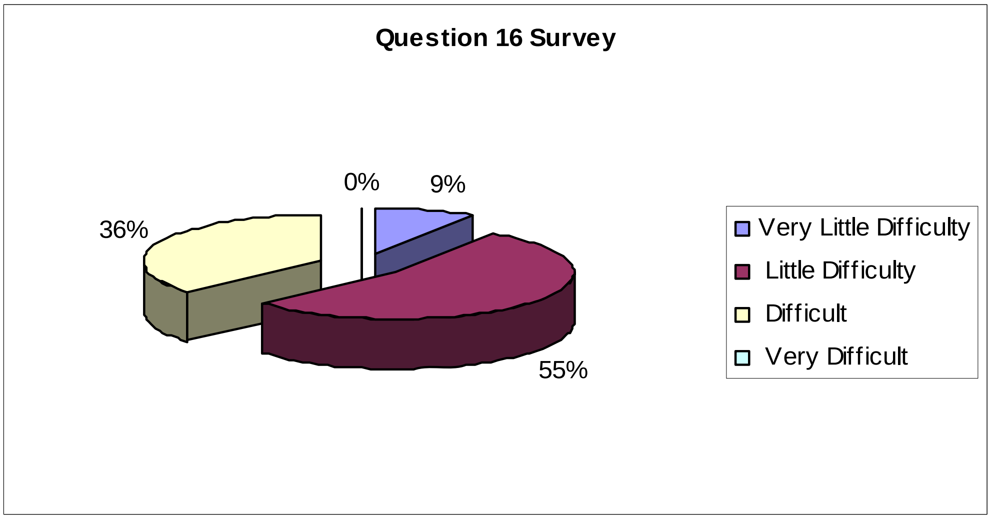

Question 10 Survey
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
Very Useful
Usefull
Not so Useful
Useless
Separation into Three Tiers

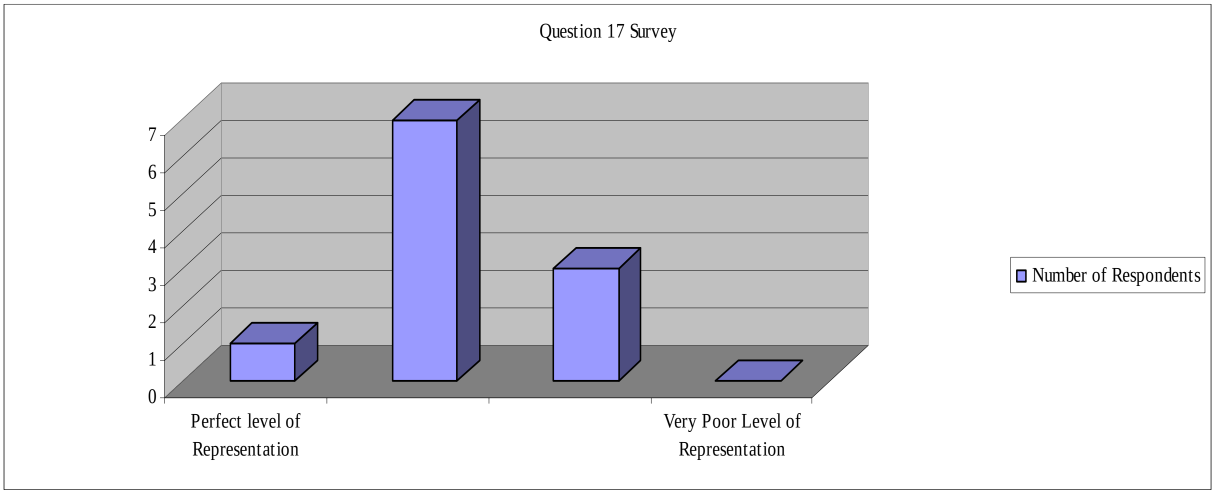

Question 11 & 12 Survey
8
6
4
2
0
Very Useful
Useful
Not so Useful
Useless
Usefulness of Activity Diagrams for System Control Flow
Usefulness of Activity Diagrams for System Data Flows

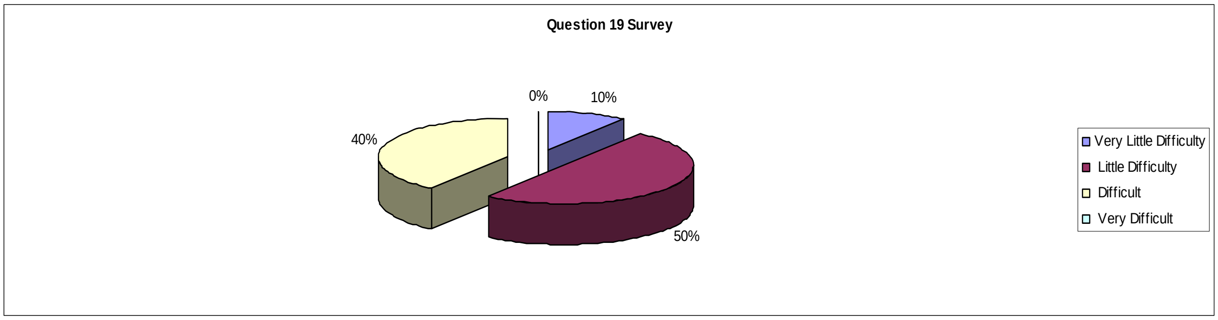

Question 13 Survey
9%
9%
18%
64%
Very Useful
Useful
Not so Useful
Useless

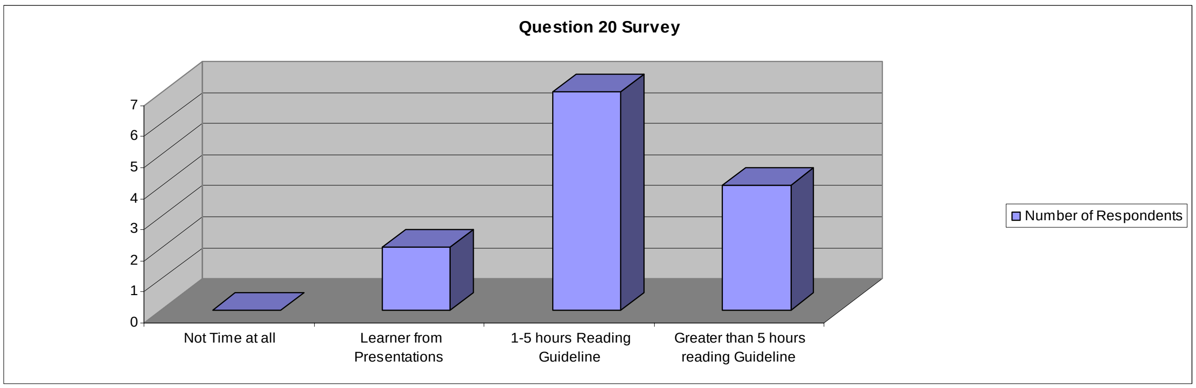

Question 14 Survey
8%
8%
33%
51%
Very Little Difficulty
Little Difficulty
Difficult
Very Difficult

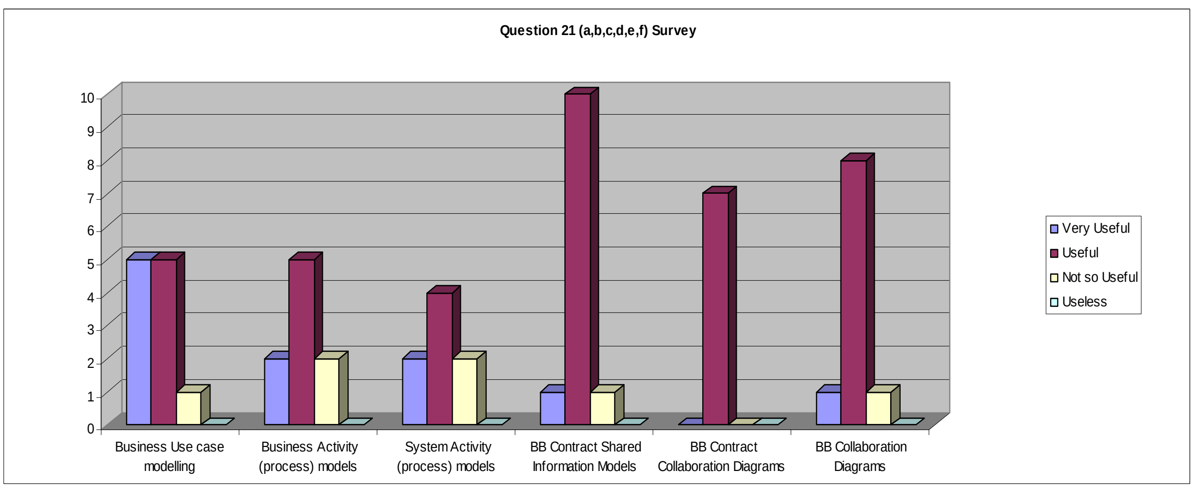

Question 15 Survey
0%
17%
50%
33%
Very Little Difficulty
Little Difficulty
Difficult
Very Difficult

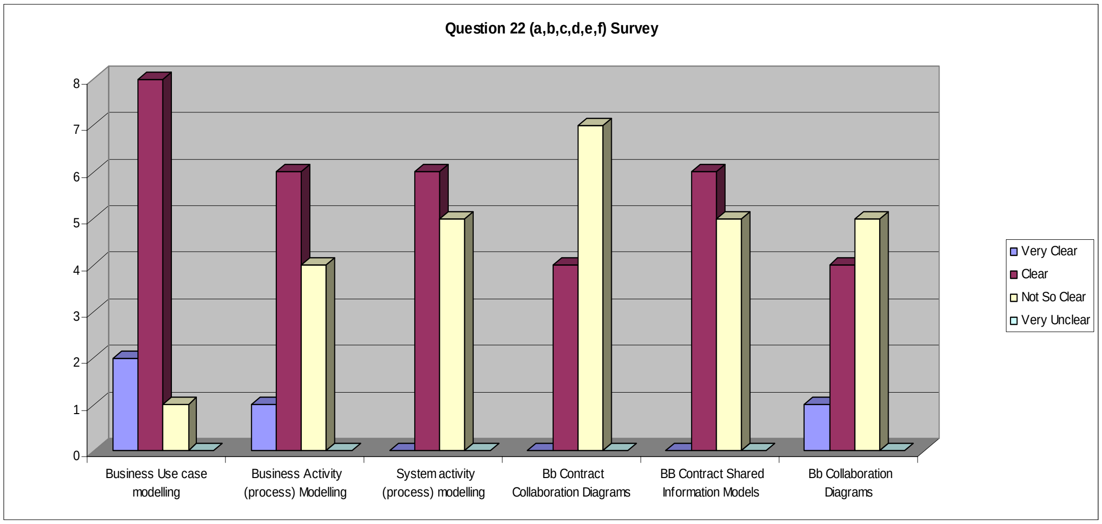

Question 16 Survey
0%
9%
36%
55%
Very Little Difficulty
Little Difficulty
Difficult
Very Difficult


Question 17 Survey
7
6
5
4
3
2
1
0
Perfect level of Representation
Very Poor Level of Representation
Number of Respondents


Question 19 Survey
0%
10%
40%
50%
Very Little Difficulty
Little Difficulty
Difficult
Very Difficult


Question 20 Survey
0
1
2
3
4
5
6
7
Not Time at all
Learner from Presentations
1-5 hours Reading Guideline
Greater than 5 hours reading Guideline
Number of Respondents


Question 21 (a,b,c,d,e,f) Survey
0
1
2
3
4
5
6
7
8
9
10
Business Use case modelling
Business Activity (process) models
System Activity (process) models
BB Contract Shared Information Models
BB Contract Collaboration Diagrams
BB Collaboration Diagrams
Very Useful
Useful
Not so Useful
Useless


Question 22 (a,b,c,d,e,f) Survey
0
1
2
3
4
5
6
7
8
Business Use case modelling
Business Activity (process) Modelling
System activity (process) modelling
Bb Contract Collaboration Diagrams
BB Contract Shared Information Models
Bb Collaboration Diagrams
Very Clear
Clear
Not So Clear
Very Unclear