University of Technology, Sydney
Faculty of Engineering and Information Technology

CARMA: Complete Autonomous Responsible Management Agent (System)

Submitted by:
Haydn Mearns
BE (Soft.)

2012

Principal Supervisor: Adjunct Prof. John Leaney
Co-supervisor: Emeritus Prof John Debenham

Submitted for the degree of
DOCTOR OF PHILOSOPHY

Certificate of Authorship/Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Acknowledgements

The research presented here would not have been possible without the help of a great number of people. I am grateful for the opportunity to be able to thank a few of them.

Primarily I am would like to thank John Leaney, my supervisor, for his tireless support through out the years, his enthusiasm and belief in this work, his guidance in all things, and most especially his patience. He has been a wonderful mentor and is most deeply appreciated.

Artem Parakhine, for his encouragement, his support, his input and guidance into the vagaries of both research and research life.

John Debenham, my co-supervisor, for his kindness and open supervision, and his determination to keep my research focused.

Alcatel-Lucent, for their financial support, and their guidance, without which this research would not have been possible.

The University of Technology, Sydney, the Faculty of Engineering and Information Technology and graduate school whose financial support made it possible to complete this research.

And finally my Family, Edwin, Robyn, and my sister Clara, whose continued support through the years could always be relied on.

Contents

List of Figures

List of Tables

Abstract

The continuing expansion of telecommunication service domains, from Quality of Service guaranteed connectivity to ubiquitous cloud environments, has introduced an ever increasing level of complexity in the field of service management. This complexity arises not only from the sheer variability in service requirements but also through the required but ill-defined interaction of multiple organisations and providers. As a result of this complexity and variability, the provisioning and performance of current services is adversely affected, often with little or no accountability to the users of the service.

This exposes a need for total coverage in the management of such complex services, a system which provides for service *responsibility*. Service responsibility is defined as the provisioning of service resilience and the judgement of service risk across all the service components. To be effective in responsible management for current complex services, any framework must be able to interact with multiple providers and management systems. The CARMA framework proposed by this thesis, aims to fulfil these requirements through a multi-agent system, that is based in a global market, and can negotiate and be responsible for multiple complex services.

The research presented in this thesis draws upon previous research in the fields of Network Management and Cloud service management, and utilises agent technology to build a system that is capable of providing resilient and risk aware management of services comprised of multiple providers. To this end the research aims to present the architecture, agent functionality and interactions of the CARMA system, as well as the structure of the marketplace, contract specification and risk management.

As the scope and concepts of the proposed system are relatively unexplored, a model and simulation were developed to verify the concepts, explore the issues, assess the assumptions and validate the system. The results of the simulation determined that the introduction of CARMA has the potential to reduce the risk in contracting new services, increase the reliability of contracted services, and increase the utility of providers participating in the market.

Chapter 1

Introduction

The continued improvement in technology over the years has led to the telecommunications environment expanding to include all facets of daily life and business. Individuals and businesses utilise network based technologies to communicate, work, shop, and interact with other companies and individuals across the globe utilising a multitude of devices from multiple vendors. This expansion has been accompanied by an increase in the complexity of the network based services being offered. From Video Conferences to virtual machines and services in a cloud computing environment, telecommunication users have access to a vast variety of services offered by multiple providers.

At the same time the non functional requirements, such as quality, of these services has become more important. Networks, originally designed under best effort technology, provisioned services by simply increasing the capacity of the network over the requirements of the service. This situation however, is no longer sufficient for guaranteeing service quality. New technologies, such as Multi-Protocol Label Switching (MPLS)(Rosen et al., 2001), and to a lesser extent Differentiated Services (Diffserv)(Blake et al., 1998), have allowed providers to provision individual services at differing levels of quality. As a consequence of this shift in management, from the provisioning of networks to the provisioning of quality individual services, telecommunication providers have been left with large over provisioned networks that are under-utilised.

As the control over individual services improves, the focus of management is moving away from simple fault resolution and towards cost and profitability. Telecommunication companies and service providers are looking to leverage their networks in a manner that maximises the profit to the company. In this environment the question for large telecommunications companies is how to leverage the capabilities of their existing networks in a more cost effective manner. For service providers the question is how to reach the greatest customer base they can to increase utilisation. For customers or companies the question is how to reduce the overall cost of their information technology infrastructure through the use of remote and dynamic services.

The full potential of the vast variety of remote services has yet to be realised however, as there are still limitations to the control that can be implemented by individual management systems. Individually, telecommunication companies and cloud providers can now exert greater control inline with quality objectives. However, the management systems have no ability to effect management on the sections of the service that are situated in a domain not in the management systems control. As an example, the cloud computing environment has allowed businesses to expand their services via the utilisation of virtual machines in a remote location, giving the business the ability to increase service ability cheaply and dynamically. This approach though has drawbacks, one of which is that a virtual machine running a program in a cloud environment is dependant on the quality of the connection from the business to the virtual machine. If the connectivity fails, then the machine is useless. Exacerbating the inter-domain issues, for users and businesses wishing to take advantage of remote services, is the inability of the user to determine the source of the problem, or correctly and quickly take action if the source is determined. Taking the same example of the virtual machine, if the virtual machine fails to respond, the business has no easy way to determine where the fault lies or who to co-operate with to get it fixed. This problem is considered to be a problem of responsibility. While this issue has been prevalent for a number of years, the increase in utilisation of Cloud computing resources, such as virtual machines has given a new prominence to the responsibility problem.

1.1 Managing Complex Telecommunication Services

Cloud computing is a relatively recent introduction to telecommunications, that offers businesses multiple advantages to traditional methods of IT. It is no longer necessary for companies to own and maintain the physical hardware required for their services, rather the providers of such services can utilise virtual representations of the physical servers to support their applications and place them in geographic locations that are advantageous for dynamic periods of time. Public clouds allow service providers to rapidly increase and decrease the availability of their service, with relatively low overhead. These advantages have led to a rapid growth in the public cloud industry and telecommunication companies are looking to provide hybrid clouds to leverage their already established infrastructure.

At the same time the domain of traditional telecommunication services is ever increasing, with the introduction of multimedia based services, such as video or audio conferences or video on demand. All of which require greater adherence to quality or performance requirements to maintain a satisfying service. In conjunction with the performance requirements, the variability of service demands, from base usage to peak usage times exacerbates the quality issues, when dealing with multiple services integration.

This increase in network capability however, has not been reflected in its management. Current management for telecommunication companies is still based on protocols and technology developed over 20 years ago. The vast majority of telecommunication companies rely on the Simple Network Management Protocol (SNMP), and the Open Systems

Interconnection model (OSI), which is generally restricted to reporting and static reactive efforts (Wallin and Leijon, 2009). In most circumstances, such reactive efforts can be quite effective, for example if a cloud providers management system detects the failure of an instance, the ability to initialise another can solve the problem, likewise if a router dies properly redundant networks can reconfigure routes without loss of service. However the effectiveness of these methods of management degrades in proportion to the complexity of the network being managed. Current telecommunication networks can incorporate thousands of devices, interacting in multiple ways. The complexity is increased as it is common for networks, as with other large computer based systems, to utilise heterogeneous devices which are purchased from multiple vendors. This has led to a problem of scalability in network management systems, as the most effective management of networks is a centralised one.

Over the past 10 years there have been multiple research projects aimed at overcoming the limitations of reactive systems and encouraging greater control based in line with company goals and policies (Section 2.2 onwards). Such management systems, from policy to autonomic based, have been focused on providing service quality based on the policies determined at the company level. This focus hinders companies ability to provide total service coverage specifically coverage of the provider interaction. These approaches have not provided the final solution and an additional overarching approach seems advisable.

1.1.1 Comprehensive Management

As the utility of Cloud computing grows, however, the composition of modern services might not lie exclusively in the telecommunication domain or the cloud, rather in a complex interaction of the two. Content servers, for a Video On Demand service, might exist in various cloud environments and rely on multiple telecommunication infrastructures to stream the requested video to the consumer. As the interaction between these two domains of telecommunications grows, so does the need for their management.

Even in the realm of current telecommunication services, where each complex service can involve multiple providers in multiple domains working in concert to provide the service. The case of more severe failures of a single provider, where such reactive systems are not effective in solving the problem. Failure of one provider requires manual intervention or re-negotiation by a team of engineers working in concert, with the accompanying time delay being undesirable for a single service, especially if that service is provided dynamically for a fixed length of time.

The *reservoir* project (Rochwerger et al., 2009) argues that two main issues of the current cloud environment is a lack of scalability in single cloud providers, and the lack of interoperability between them. We would add a further issue to these, namely responsibility for the entire complex service. For responsible coverage of any secure video processing application run on a public cloud, there also needs some secure quality assured bandwidth between the user and the application and its components. The management of which is just as important as the management of the virtual machine running in the cloud.

1.1.2 Dynamic Market for Service Fulfilment

In order to deliver current complex services in any kind of dynamic way, telecommunication and cloud providers need to be able to purchase services quickly and efficiently. Like the current methods of dealing with failure in a complex service, current methods of negotiating service level agreements with partnering providers is generally an offline process, requiring a significant lead time to agree on service criteria such as performance parameters.

This is a significant limitation when new services are required quickly or on demand. Current research envisions (Vaughan-Nichols, 2011; Lefevre, 2005), that with the increase in technological capability dynamic network services can be provided to match current cloud based services. Such services, as well as more basic services, (such as guaranteed QoS for network connectivity(Turner et al., 2010)), would be offered in an open market place, negotiating service level agreements to agree on duration, price and quality.

1.2 Motivation

Modern telecommunications require multiple telecommunication network management systems to provide complex services for users. Telecommunication management systems have no provision to deal with the problems occurring in the interaction of multiple providers. The purpose of this thesis is to address the limitations of current management systems by firstly, aiming to give coverage of the total service sought in the interaction of telecommunications and the cloud, and secondly, to take responsibility for that service.

- **Coverage** of the total end-to-end service means that the cloud service is to be seen as the collection of all the user's applications and computing services, together with their related communications services.
- **Responsibility** for the service is to be responsible for the coverage, i.e., both the applications and the communications. It is further defined as providing risk management for the services (to price fairly and manage effectively), to negotiate, provision and schedule, and, to ensure a resilient service (in which poor performance and total failure are managed).
    - **Risk Management** What is needed is something that can ultimately accept the risk of providing the total complex service, by managing the risk of multiple single service providers, for multiple services, accepting the risk for such services and guaranteeing delivery.

  - **Resilience** In order to guarantee the delivery the system must have the ability to manage the performance of the complex service, with the ability to initiate any re-configuration of the complex service where necessary.

1.2.1 Vision

A marketplace in which all manner of complex services will be provisioned, and their performance managed across multiple domains, accepting the responsibility of each service. The approach of this thesis is to develop an appropriate service model, based on negotiation. This thesis is using the terminology of Bundled Services as a representative end-to-end complex service.

1.2.2 Research Questions

The motivation and vision of this thesis engenders a number of questions which were then used to guide the research.

1. In a multiple provider environment, what are the provider independent properties of a dynamic complex service?
2. What is the current state of service coverage in multiple provider complex service management?
3. What does it mean to be Responsible for a service? Specifically with regards to Resilience:
    (a) How can the performance, in terms of Quality of Service, of a service involving multiple independent providers be judged?
    (b) In the context of multiple service providers, working in concert to provide a dynamically created complex service, how can service resilience, be ensured?
4. With regards to Risk Management:
    (a) How can the risk of contracting a service be managed, when the service involves multiple individual providers?
    (b) How can the risk of maintaining the service under poor performance be judged?
5. What kind of framework can provide resilience, risk management, and total coverage of complex services involving multiple independent providers?
6. What other elements are required by the framework to ensure effective responsible coverage of dynamic services in the multi-provider environment?
7. How can the existing concepts of market operation assist in responsible service?

1.3 Research Methodology

Due to the nature of the problem, the skills of the candidate, and the nature of the research this was always going to be an experimental thesis based in simulation. As such the methodology chosen for this research is Action Research (AR). AR is a cyclic methodology which is a research approach analogous to iterative development and provides an emergent process that

> takes shape as understanding increases. (Dick, 1999)

As such it subscribes to the Constructionist epistemology, in saying that the truth (or in this case solution) is not objective. Instead it specifies that whatever meaning that exists in the research is 'constructed' from the interaction of the researcher and environment. Interactions, or just actions, form an important part of any design process and reflect the researches objective of designing and building an architecture. AR is also defined by its view of critical reflection periods, both preceding action tasks and following them. This cyclic method has been described as

> plan, act, observe, reflect, then plan again(Kemmis and McTaggart, 1988)

shown in Figure 1.1. Since the improvements and solutions proposed will emerge from the unique analysis of current management research and industry practice, the research methodology used needs to be flexible in managing less clearly defined objectives. In AR the results of each cycle will influence the planning of the next cycle, refining and expanding on the positive aspects of the outcomes and re-evaluating the negatives. This research method will involve modelling techniques, both computer based and formal, to test the network management architectures and any proposed improvements. The research also needs to take into account the practicality of such systems with regards to current industry practice.

The focus of this thesis is the design of a system which fulfils the requirements as stated in the motivation and expressed through the research questions. This will be accomplished through simulation. Simulation was chosen over a closed form analytical model as the motivation for, and the questions posed by this thesis involve the interaction of multiple entities, such as Cloud computing providers and telecommunication providers. This interaction is proposed through the use of a multi-agent system, the core of which is interactions that can change as circumstances in the environment change, with the desire that these changing circumstances reflect real world conditions and scenarios (Wooldridge, 2009, p.248). Further the non-functional performance related goals of the thesis, such as scalability and resilience as a component of quality, are impractical to model in a closed form.

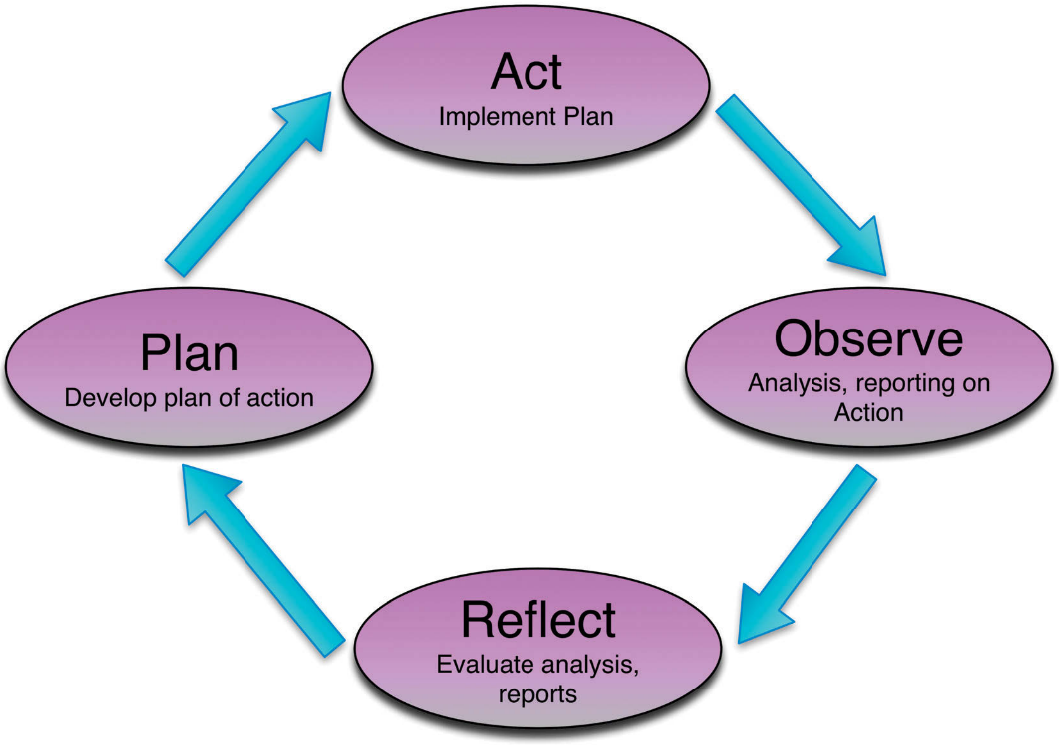


Figure 1.1: Action Research cycle

1.3.1 Plan

The general structure of a plan in the AR cycle will be formed by the information gathered in previous cycles, current research in the academic literature and the current state of the industry. Planning is a very dynamic, utilising intuition, experimentation, and has been described as

> conversing with the problem (Schön, 1983)

This incremental approach, encompasses many small steps and changes. The changes themselves influence the planner, adjusting the plan and allowing a greater understanding of the problem. Unfortunately, this approach can seem poorly structured for academic researchers as it contains fluid goals and evaluation criteria. In order to counteract that perception, the methodology of Reflection in Action (Schön, 1983) will be used. Reflection in Action stresses the importance of explicit and repeated reflection after each small change. While this still leaves the goals and criteria of the reflections themselves relatively opaque, relying on the skill and intuition of the practitioner, the experimental process becomes much more rigorous. For this thesis the definition of the planning cycle is focused on the design and redesign of the system formulated as a proposed solution to the motivation and questions raised in this introduction.

1.3.2 Act

The importance of testing the validity of telecommunication management frameworks, as well as later improvements and changes, requires some understanding of the current management systems performance. I therefore initially researched current telecommunication and Cloud computing management and proposals then later simulated the performance of current management systems proposals as well my own design. This simulation focuses on the performance characteristics of the proposed design, specifically focusing on scalability, responsibility, and effectiveness of the system. The final cycle will involve the validation and verification testing (VVT) of the proposed solution to the stated objectives. In order to give some rigour to these decisions, Reflection in Action (Schön, 1983) will be used. The action in this thesis was primarily accomplished though simulation designs and changes, and the contract specification and transformation.

1.3.3 Observe and Reflect

The observation phases involves analysis of the results of the simulation as well as analysis of the strengths and weaknesses of the architectures, both those currently proposed and my own, via the simulations built during each act phase or cycle. Further activities during these phases involved the writing of these findings for conference papers and journals, and further research in the problem space with respect to new developments.

The Reflection phase can be defined as the most important in the AR cycle, providing the needed rigour and validity from an academic standpoint. With regular, critical and systematic reflection, there is more confidence in the research conclusions (Dick, 2002). In AR in particular, the focus on reflection encourages the inclusion of evidence that does not fit with what is expected and assumed. Although this reflection is specified in the methodology as a specific phase, reflection can and does occur during all phases of the AR cycle. These smaller moments of action and reflection have been described as

> cycles whithin cycles within cycles (Dick, 2000)

As they reflect the understanding of practitioners that AR cycles can occur over a variety of time spans, from the entire thesis down to moment by moment decisions (Dick, 2000). In every seeming unique problem approached in the plan phases there is reflection on the solution proposed. This will lead to changes in planning as the implications of the reflection are considered. Similarly in the act phases, each of the myriad design and simulation problems will lead to reflection on the desired goals and a re-factoring of the design or simulation. Finally during the Observation and Reflection phases the results of the analysis and testing led to new possibilities or problems which needed to be addressed.

1.3.4 Implementation

For this thesis the implementation of the method was used lightly and tied to the simulation and contract specification development as the methodology of action research maintains a close relationship to the software development methodology of iterative development. However, the structure of this thesis was written in a linear manner to reflect the nature of an experimental thesis in an engineering environment. In this thesis the AR research was divided into three cycles. For clarity the three cycles are described below and the relevant sections, related to the planning, action, observation and reflection is listed.

1.3.4.1 First Cycle

The First Cycle of the AR research was primarily focused on investigation of current management services, and analysis of the management systems performance in the context of the motivation and research questions. Secondarily, there was an investigation into the historical and current state of telecommunication and cloud computing management in practice, as related to management architectures and the performance of the systems. The planning for both objectives in this cycle formed the basis for the literature review, and the results of the observation and reflection parts of the cycle are detailed in Chapter 2. Specifically, for the primary focus, with regards to the management system investigation and analysis is covered in Section 2.2, Section 2.4 and Section 2.5.1. Analysis of the performance of management systems is further presented in the publication of the author (Mearns et al., 2010b). For the secondary focus on the historical and current management practice, the Sections 2.1 and 2.7 cover the observations and reflections that resulted. Additionally, a second paper by the author (Mearns et al., 2010a) expands on the reflection detailing the architectural evolution of network management.

1.3.4.2 Second Cycle

The Second Cycle involved addressing the limitations in the context of the motivation and research questions. This cycle was focused on designing the architecture for a multi-agent system, that could address the issues and limitations that were identified in the first cycle and specified in Chapter 2. For this cycle the main goals were defining, the requirements of the system including the entities of the proposed system, the requirements for the market place environment that the system operates in, an initial architecture of the system and an initial design. The planning in this cycle involved defining the requirements of the system, which is described in Section 3.1, defining the requirements of the marketplace in Section 3.4, and defining the requirements of risk management in Section 3.5.2. The action portion of this cycle involved the specification of the entities described in Section 3.2, the specification of the architecture for this system described in Section 3.5, the scope of the contract specification described in Section 3.3, and an initial model and simulation, described in Section 5.4.1. The observations and reflections of this section involved the understanding of the limitations of petri-net design as implemented in the chosen modelling framework, also described in 5.4.1, and reflections on the architecture as shown by the initial simulation which informed on the design in Chapter 4.

1.3.4.3 Third Cycle

The third cycle was concerned with the final design of the system with regards to addressing the issues and requirements discovered in the second cycle both with the functionality of the design and simulation. The planning for this cycle involved determining an appropriate marketplace for the services, which is described in Section 3.4, determining and appropriate method of risk management, described in Section 4.4. Defining appropriate contract specification and transformation methodology specified in Sections 4.3.1 and 4.3.2. Further planning involved the definition of entity and agent functionalities and interactions, which is described in Section 4.2 and building the appropriate structures in the simulation which is covered in Section 5.4. The main action of this cycle was verification and validation testing of the simulation based on the case study the description of which covers Chapter 6. As appropriate to the action research concept of cycles within cycles, this action phase involved multiple mini cycles of development and reflection, which is covered in the Chapter. The observations and reflections on this cycle is described in the discussion of Chapter 6 and conclusion in Chapter 7.

1.4 Thesis Structure

There was much reflection as to whether or not the structure of the thesis should reflect the structure of the AR cycles or be presented as a conventional computing thesis. It was decided to present it as a conventional computing thesis, because the AR cycles did tend to be concerned with requirements or design or implementation. So it was simpler to present it this way.

Chapter 1 The current chapter. This chapter is concerned with introducing the environment of the perceived problem, as well as the premise that led to the framework, ideas and concepts upon which this thesis is based.

Chapter 2 Provides an overview of the historical and current practice in the telecommunication and cloud management environment as well as the relevant literature designed to improve the management of individual networks and cloud environments.

Chapter 3 Provides the framework for the proposed system, detailing the requirements for risk management, contract specification, market requirments and definition, entities and their relationships, and the system architecture.

Chapter 4 This chapter is concerned with an in-depth discussion of the management system, structure, agent interaction, contract specification and transformation and risk management proposal.

Chapter 5 Specifies the simulation designed for the analysis of the proposed system, including agent structure, resource usage simulation, market implementation, risk management implementation and a description of the simulation case study.

Chapter 6 Describes the results of the simulation, specifically provides a verification of the simulations correctness with regards to the system design, and a validation of the design with regards to the stated goals. Included in this chapter is the further refinements to the design that were predicated by analysis of previous simulation runs and discussion on the assumptions and issues discovered.

Chapter 7 Contains the summary and conclusions of this research detailed throughout the chapters of this thesis.

1.5 Publications

Mearns, H., Leaney, J. and Verchere, D. (2009). The architectural evolution and performance bottlenecks of telecommunications network management systems, *Proceedings of the 17th IEEE International Conference and Workshops on Engineering of Computer-Based Systems*, IEEE, pp. 281-285.

Mearns, H., Leaney, J. and Verchere, D. (2010). Critique of network management systems and their practicality, *Engineering of Autonomic and Autonomous Systems (EASe), 2010 Seventh IEEE International Conference and Workshops on*, IEEE, pp. 51-59.

Mearns, H., Leaney, J., Parakhine, A., Debenham, J. and Verchere, D. (2011), An autonomic open marketplace for service management and resilience, *Network and Service Management (CNSM), 2011 7th International Conference on*, IEEE , pp. 1-5.

Mearns, H., Leaney, J., Parakhine, A., Debenham, J., and Verchere, D. (2011), An Autonomic Open Marketplace for Inter-Cloud Service Management, *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, pp. 186-193,

Mearns, H., Leaney, J., Parakhine, A., Debenham, J. and Verchere, D., (2012). CARMA: Complete autonomous responsible management agents for telecommunications and inter-cloud services, *Network Operations and Management Symposium (NOMS)*, IEEE , pp.1089-1095.

Chapter 2

Literature Review

The aim of this chapter is to review the state of the art with regards to the problem of managing telecommunication based services across multiple domains. This problem has arisen as the focus of network management has changed over the years. With the introduction of the Internet Protocol (IP) and its basis in best effort connectivity, network management was focused primarily on the optimisation of the whole network, with no concern for ‘quality of service’ for individual services. In the late 90’s a service centric view was introduced, with the focus being on aligning the management of network services with the policies of the network owner. This was driven by the increase in utility of network based services. This increase in utility meant that businesses were looking to leverage their network presence into economic value. Research into network management in the latter half of this decade (the 2000s, Section 2.4) has been focused on integrating this policy centric view with the idea of autonomic control for networks and network devices. The purpose of this autonomic focus is to provide the network with the flexibility to adapt to changing circumstance in the network by analysing the situation and applying the most appropriate policy for the situation. The other half of the autonomic principle is the move away from centralised network management systems, with each autonomic agent providing independent control for the devices it manages under the previously mentioned policy guidelines.

The business focus of network management systems has introduced further limitations in terms of scalability, as centralised management approaches struggle to accommodate the ever increasing heterogeneous devices in the network. Agents, being inherently decentralised, provide an interesting solution for management systems, with the more recent management proposals such as the autonomic focus utilising agent systems to distribute management systems.

Current network management in practice is concerned with managing the network from the perspective of the domain owner or business entity. However, current telecommunication services utilise the services of multiple providers to complete the service, with the result that there is no total coverage for a single service. There are valid reasons for this disconnect from the companies point of view, such as security, competitive advantage, and the sheer impracticability of combining multiple network and service management systems that are owned by different companies. This lack of comprehensive monitoring and control over all elements of a particular service results in a lack of responsibility to the user of the service. This lack of responsibility is particularly frustrating to users of any composite modern telecommunication service, for if a service fails, tracing the cause of that failure results in a significant time delay.

For comprehensive management systems to deal effectively with the multiple providers with which it has to interact, there needs to be a environment that is conducive to that interaction. To date there are few methods which specifically target the problems that arise with the integration of composite services. There have been approaches to integrate multiple cloud computing environments which will be described below. These approaches follow one main method, that of federation, with and without market involvement. However these approaches focus on the problem of scalability in cloud environments and ignore the telecommunication providers in between the services.

The other approach is that of utilising market forces to provide the required interaction and pricing for building composite services. The research into the various types of market has been extensive and researchers have been investigating the most effective form of market interaction. From auction based initiatives to negotiation and exchange style markets.

For background, this review initially covers the general network management history along with the changes in network and remote services technology and the effect that has had on traditional network management. Once the traditional state of network management has been established, the review will evaluate the current state of telecommunication management in industry, evaluating the popular utilisation of common telecommunication and cloud computing management systems through the focus of responsibility to individual service management. This chapter will then present a survey of current methods for improving network management, as well as Cloud Computing management, and the integration of multiple domains through markets.

2.1 History of Network Management

The concept of network management has been around as long as telephony. Indeed the first telephone operators could be considered to be network managers. However it was the introduction of the distributed computing environment that expanded the view of network management to include systems of

> diverse types and sizes from multiple vendors (Voruganti, 1994)

Networks became dynamic, supporting a variety of applications, and consist of thousands of subsystems and devices. Traditionally the management of these devices has been managed through manual efforts supported by standards and protocols, such as Simple

Network Management Protocol (SNMP) (Case et al., 1990), and OSI (LaBarre, 1991), created over 20 years ago.

The SNMP management model employs three components, the manager or client, the agent or server, and the Management Information Base (MIB). The MIB is created by the vendor of each network device, and to be effective, the manager or client needs to be able to access the MIB for each device. Similarly the OSI utilises the client/server (manager/agent) model, but defines a greater number of message types than SNMP, through the definition of five distinct management areas. The areas are defined as Fault management, Accounting management, Configuration and name management, Performance management, and Security management (FCAPS). OSI was one of the first attempts to build comprehensive management that did not just focus on faults, but attempted to address the other realms of network management.

As management matured SNMP dominated the market with devices being sold with SNMP based agents and accompanying MIBs. The SNMP model, with its client server architecture, conformed to the traditional idea of network management structure, which was to utilise a centralised *platform* framework(Kahani and Beadle, 1997), with management applications being separated from the monitoring data and from the devises that are under the applications control. This traditional centralised view of management treated the companies network as a whole and concentrated on fault management. As the diversity of services increased, this traditional view became problematic as the complexity of service interaction highlighted problems of scalability, traceability, and flexibility in the centralised management approach. Research into network management, moved focus on a service centric view, that is, rather than just fault management of individual devices performance data was analysed based on the Quality of Service (QoS) ideal, and devices managed on a service by service basis.

2.2 Policy based Network Management Systems

At this time there was a growing interest in providing management solutions that can manage the low level decision making in the provisioning or adjusting of a new service. In the late 90's one of the first ideas to create this automatic decision process was the idea of Policy Based Network Management (PBNM). Policy based management is defined as

> the use of policy rules to manage the configuration and behaviour of one or more entities (Strassner, 2003, p.56)

Policy management works through an event driven architecture, utilising the *Event-Condition-Action* (ECA) structure. Strassner goes on to describe early PBNM as

> characterised as a sophisticated way to manipulate different types of QoS (Strassner, 2003)

However by 2003 there was a growing understanding that policies needed to be integrated with business rules and processes. This created the need for *abstraction* when dealing with more sophisticated concepts, such as differentiated services, users and resources. This led to the business rules of the service provider becoming the authority for changing QoS requirements. A formalisation of this concept can be seen in Strassner's Policy Continuum (Strassner, 2003, p.367), which exposes 5 layers of policy from the high level Business view, to the low level Instance view, and can be seen in Figure 2.1. The specification of the Policy based management has been implemented in a variety of Domain Specific languages such as the early PDL (Koch et al., 1996), Ponder(Lymberopoulos et al., 2003), PRONTO (Sheridan-Smith et al., 2006) and Strassner's own DEN-ng (Strassner, 2003) that specify the set of classes and relationships, that represent the semantics of the building blocks of policies.

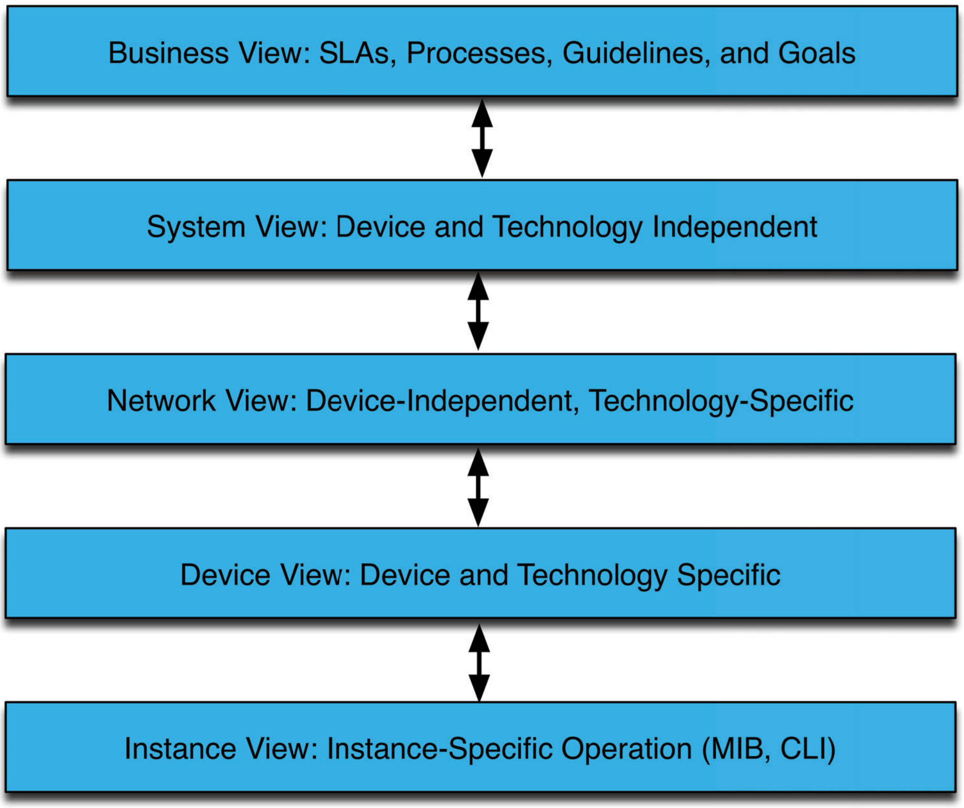


Figure 2.1: Strassners Policy Continuum.

While the use of PBNM increases the capabilities of the network management system to dynamically configure devices, the structure of policy based management is still relatively centralised. Through the use of both Policy Decision Points(PDP), Policy Enforcement Points(PEP), and the Policy Continuum, PBNM implements a hierarchical structure, with low level decisions being made by PDP's in the Instance view, and higher level decisions being made by the PDP's in the higher levels to the Business view. One consistent problem with PBNM is conflicts that arise through the implementation of different and concurrent policies, often at different levels. This conflict can result in poorly configured devices and services, which can lead to poor service performance and failure. Initial attempts for conflict resolution required a centralised policy model, housed in the Policy management tool, limiting the scalability of the management system despite the hierarchical distribution of the PDP's and PEPs. Further the centralised policy model is impractical when dealing with cross-domain networking. More recently through the use of ontologies, most commonly using the Web Ontology Language (OWL) (*OWL Web Ontology Language*, 2010), PBNM has tried to address the policy conflict problems.

2.3 Service Management

As Policy based Network Management was emerging, researchers were already looking at the integration of the telecommunication and information technology industries. The TINA Consortium (Inoue et al., 1998) was an attempt by 40 leading telecommunication vendors, network operators and IT vendors to create a ”cooperative solution to a competitive world”, and supported (amongst other things) Openness, Support of new services, and Uniform support of management. The goal of the TINA architecture was the provisioning of any service through different network providers and infrastructure, on a global scale. While TINA was concerned with Quality of Service requirements for its provisioned services, it did not employ policy models, and defined its own similar views, through the Reference Model of Open Distributed Processing (RM-ODP).

The TINA architecture hoped to use standard interfaces (implemented in CORBA), to negotiate global services through its Distributed Processing Environment(DPE). The collection of DPE's would create logical separated networks (kTN), regulated by the TINA session model. Unfortunately this system was difficult to implement with the technologies of the time, with QoS technologies still in their infancy, and it never found the cooperation by telecommunication and IT providers needed to get past the initial phase of an architectural framework, component specification and a feasibility study.

In 2000 another consortium of telecommunication vendors, started working on the implementation of intra and inter domain QoS based services. The TEQUILA consortium's goal was to provide

> A validated framework for the provisioning/definition of end-to-end Quality of Service(QoS) though the internet (Mykoniati et al., 2003)

Unlike TINA, TEQUILA focused on traffic engineering, the specific process of establishing how traffic is treated within a given network. The TEQUILA project was influenced by the emerging technologies of the time, that being the Integrated Services (IntServ) and Differentiated Services (DiffServ) QoS technologies, which enforce the QoS requirements at the edges of a particular network. This focus on network edges, encouraged the idea of Per Hop Behaviours, where each Traffic Management Block (router to router, but could be expanded to provider or network) ensures the QoS for their block. Providers and consumers interact via Service Level Specifications, which specify the required level of quality for each service. TEQUILA also employs a policy framework for admission control policies, and proposed an extension for dynamic resource management policies.

While TEQUILA was comprehensive in its management of the intra-network, its proposals for the Inter-network relied on particular QoS expansions to BGP, which never eventuated in practice, though they have been developed in literature quite extensively. At the same time the Service Level Specifications are focused on the minutia of QoS specification, and was aimed entirely at network operators who could specify the required bandwidth jitter and delay. Finally the architecture did not specify a mechanism for the pricing of services between users and providers.

2.4 Autonomic Network Management

In 2001 IBM wrote about their vision of Autonomic Computing. Autonomic Computing uses deliberate biological connotations with Kephart and Chess describing it as a computing systems ability to:

> manage themselves given high level objectives(Kephart and Chess, 2003).

The improvement to PBNM systems is the idea inherent in the word autonomic, taken from the example of the autonomic nervous system which regulates heart rate and other low level tasks without requiring concious thought. The goal of autonomic management is Self-Management which has four Self-* properties:

- Self-Configuration, which is the ability of the devices to configure themselves to high level policies.
- Self-Optimisation, which continually seeks to improve their own performance.
- Self-Healing, which focuses on detection, diagnosis and repair of local problems and faults.
- Self-protection, in which the system defends itself against malicious attacks or detected cascading failures.

It proposes that this will be accomplished through autonomous elements, when in meeting the goal of self-management all elements have the independent ability to Monitor the current performance of the element, Analyse its performance, Plan any changes required to meet an optimal goal, and Execute those configuration changes (MAPE) in an autonomous control loop. This autonomous element and its control loop can be seen in Figure 2.2.

While the autonomic computing concept was devised to cover the management of all computing systems, it is particularly appropriate for networking. Unlike more traditional realms of computing management, network management has the advantage of a long tradition of monitoring, with the information provided being very useful in determining the situational awareness needed by an autonomous element. The knowledge requirement

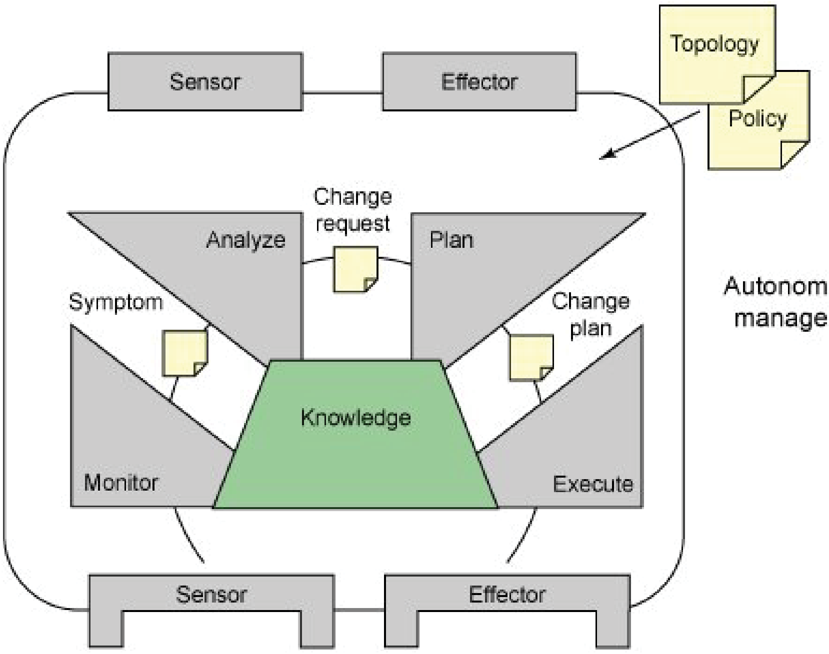


Figure 2.2: Autonomic Element with MAPE control structure.

however, is the most problematic element of autonomic networking. While local information is relatively easy to acquire, information that is required for analysis of more complex services, that of Inter-domain services for example, requires a greater range of knowledge. For global end-to-end connections, a solution that has been presented is the idea of a *knowledge plane* (Clark et al., 2003), a unified approach to aggregate global information, requirements, constraints, and goals. As yet though there has been no agreement on the implementation of this knowledge plane (Dobson et al., 2010). It is doubtful that it will rely on traditional algorithmic approaches, rather, given the scope and complexities involved, approaches based on Artificial Intelligence (AI) or cognitive systems are being considered.

For telecommunication company level networks, the solution presented by research has generally included the abstract policy models that have been previously described. The collaboration between policy based network management and autonomic networking is continuing in projects like FOCALE (Jennings et al., 2007), ANEMA (Derbel et al., 2009), and NetServ NAME (Femminella et al., 2011) which will be described in more detail below. However, previous work done by the author, has described that this reliance on the centralised policy model to provide the knowledge required by the autonomous elements, becomes a performance bottleneck when dealing with large networks or domains (Mearns et al., 2010b).

2.4.1 FOCALE

Modern research into network management systems has expanded on and integrated the concepts of policy based management and autonomic networking. One large project that has been continuing for a number of years is the Foundation Observation Comparison Action Learn rEason (FOCALE) architecture (Jennings et al., 2007). It describes the design as split into a hierarchical distributed design, with the base element being a AME (autonomic management element), which handles a managed resource, be it single device or network. This AME controls the functionality of the managed resource by marrying an Autonomic Manager (AM), with a Model Based Translation Layer (MBTL), which translates the vendor specic data and commands to the AMs vendor neutral commands. The AME is contained in a Autonomic Management Domain, and Autonomic Management Environment with each layer providing context, discovery, security, policy, and analysis services. For the practical implementation of the AME, FOCALE utilises combination of information models (DEN-ng), ontologies (OWL), and Domain specific languages to derive the context model which represents the current state of the network and services. In its own parlance it does this by dividing the Monitor, Analyse, Plan, Execute (MAPE) control loop, into two, a maintenance control loop and an adjustment control loop.

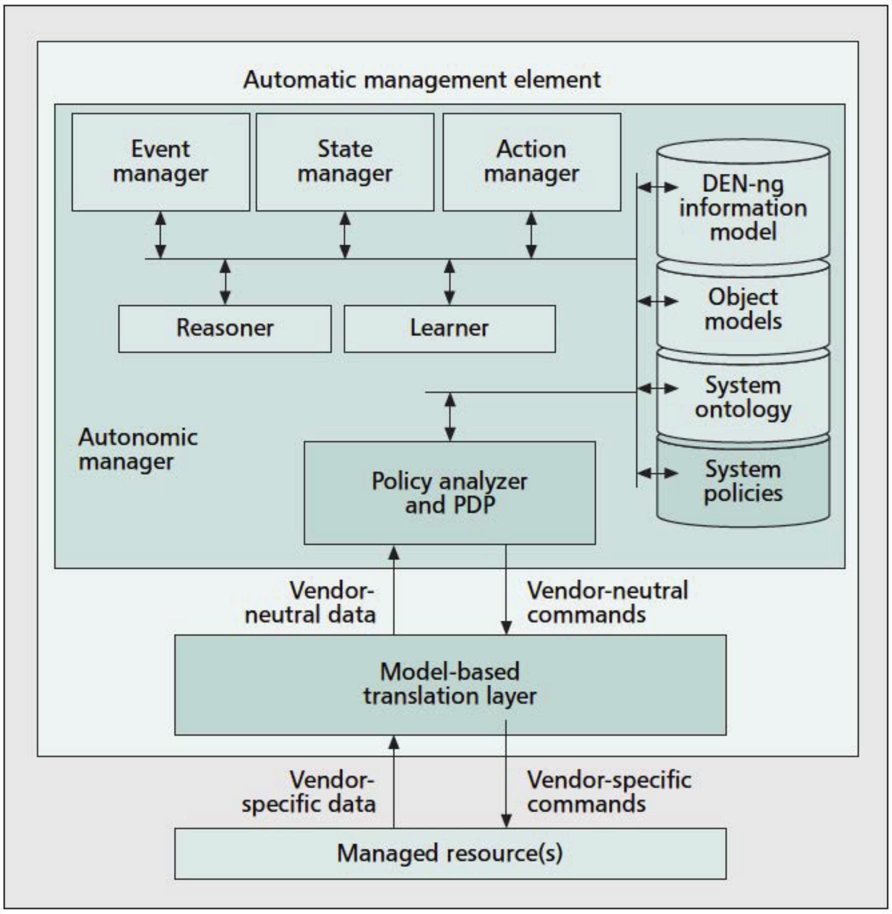


Figure 2.3: FOCALE's AME with knowledge plane and information model. (Jennings et al., 2007)

Since its conception, FOCALE has been expanded to include concepts such as biological inspired policy based management (Balasubramaniam et al., 2006), and the concept of seamless mobility (Strassner, 2009), which allows user data to be accessed regardless of the type of the device being used. Further attempts to address scalability issues in autonomic management have proposed the concept of hierarchical autonomic management (Famaey et al., 2010). Hierarchical autonomic management defined the use of child and parent AME's for use at the managed element, data-centre, organisational and inter-organisational layers. However, while the benefits of the hierarchical architecture are explained and tied to the policy continuum, the inter-organisational element is not fully explored.

There have also been greater attempts at refinement through policy conflict algorithms that utilise the policy continuum (Davy et al., 2008). While all of these expansions and refinements greatly enhance the capabilities of the FOCALE architecture, it is still limited by the knowledge that it possesses of its environment. As the control required by the networks expands to a global perspective the quantity of information required by each autonomic element for informed policy decisions becomes prohibitive.

2.4.2 ANEMA

The Autonomic Network Management Architecture (ANEMA) proposed by (Derbel et al., 2009) describes an architecture which utilizes high level strategies to define goal policies to configure network elements. The high level strategies are implemented by the Objective Definition Point (ODP) component, using expert defined analytical optimisation models that express the network functionality in terms of Network Utility Functions (NUF). The NUF's are forwarded to the Goal Definition Point (GDP) component which selects appropriate management strategies, specifically the configuration and optimization strategies with which to optimise the NUF. These strategies represent goal policies that are defined as an aggregation of management strategies which are needed to achieve one or more quality metrics related to the NUF. These goal policies are distributed to the Distributed Goal Definition Points (DGDP) and analysed to identify expert given behavioural policies and rules that can be distributed to the base level of the architecture, the Autonomic Management Element (AME). Architecturally this AME is similar in structure to the FOCALE architecture.

2.4.3 Software Defined Networking

One recent advance in research that has the potential to improve the ability of management systems to pro-actively control individual networks is Sotware Defined Networking (SDN). Software Defined Networking separates the control plane (the controlling logic) from the data plane (the packet forwarding). This allows the increasingly complex logic of management to be integrated more flexibly into the network, overcoming the difficulties and misconfigurations that arise with static command line interfaces on current devices (Kim and Feamster, 2013).

2.4.3.1 NetServ NAME

The Network Autonomic Management Element (NAME) (Lee et al., 2011) is a management system that is designed to work with a virtualised network service called NetServ. NetServ utilises the Openflow (Vaughan-Nichols, 2011) SDN. NetServ NAME is based almost entirely on the previously described FOCALE architecture, with an additional Policy Employment Point (PEP) module that is designed to include the additional functionality of the NetServ programmable node architecture. The autonomic management element is designed to be deployed as a service in the managed NetServ nodes.

2.4.4 Autonomic Systems For Cloud Environments

Cloud Computing research is also looking into utilising autonomic elements. The use of autonomic behaviour is of particular interest when dealing with the dynamic nature of cloud management. Specifically in the area of resource provisioning (Goscinski and Brock, 2010; Lorimer and Sterritt, 2012), workflow management (Kim and Parashar, 2011; Ranjan et al., 2012). In both resource provisioning and workflow management the use of autonomic behaviour is seen as beneficial to the goal of efficiency in the cloud environment as the self-* properties, would create some automated optimisation in the scaling functionality of cloud applications.

2.4.5 Federated Autonomic Management

The autonomic management systems just described were designed for the management of individual networks and services provided by an individual organisation. As such they are not designed to manage services that cross domain boundaries. Conceptually there has been some acknowledgement of the requirement for management strategies that do concern multi-domain services. In particular the concept of a Federation of Autonomic management systems has been proposed and the challenges listed(Jennings et al., 2009; Serrano et al., 2010). In this context the concept of federation is described as a

> persistent organisational agreement that enables multiple autonomous entities to share capabilities in a controlled way (Agoulmine, 2010, p.103).

Implied in this definition is the key point that the entities involved contain the controlling authority on decisions made with regards to the entities resources. Jennings et. al, describe a layered federal conceptual model which covers their view of the more important aspects of the dependencies in a federation (Agoulmine, 2010, p.105).

From the perspective of this thesis the importance of this Layered Federal Model (LFM) is that it acknowledges that in order for federation to be effective, and by extension any end-to-end service management, the members of the federation must have some form of shared semantics for effective information sharing, and that the members must cooperate with regards to monitoring and auditing of service performance. However, this version of federation argues for persistence, beyond the life cycle of individual services, in effect granting access and control over remote members resources for an indeterminate amount of time. While this approach benefits the coordinator of this federation, it limits the flexibility of the coordinated members.

2.5 Cloud Management

While the first half of this review has focused on general telecommunication service providers without too much specialisation, it is worth understanding in particular the status of management for public and hybrid cloud infrastructures. With the rise in utility of cloud computing infrastructures, it is believable that the requirements of guaranteed end-to-end service quality would be driven by the needs of public and hybrid cloud users. Already researchers are looking at a future when the needs of businesses in a global market would not be satisfied with the use of one cloud provider and its geographical location. The research into the use of multiple clouds in multiple domains has been termed Inter-Cloud research. However this research is generally focussed at the interaction of Cloud-providers, while ignoring the reality that modern remote services rely on the telecommunication providers that comprise the connectivity between them.

2.5.1 Inter - Cloud Systems research

The main concept which is the focus of current inter-cloud research is that of federation. Federation aims to combine the utility of multiple cloud providers, through the use of a system of active agents that manage services with respect to the business or user, and promote interoperability between the disparate cloud providers. Federation is driven by researchers foreseeing a future lack of scalability in cloud provided services.

2.5.1.1 Reservoir

The *reservoir* federated cloud (Rochwerger et al., 2009) is an architecture for federation through *reservoir sites* with applications and resources being defined through the reservoir Service Manifest SLA's and service management agents, in order to manage the service across the providers. The architecture of reservoir is shown in Figure 2.4 and describes three main entities at different levels of abstraction. The highest level describes the Service Manager which interacts with the Virtual Execution Environment Manager (VEEM) and outside service providers. A key point of this reservoir environment is that the management of the federation is focused on the lower level *infrastructure* providers, rather than the service providers. The Service Manager receives the service manifest SLA's from the service providers and handles the negotiation, pricing and billing. It's main responsibilities is provisioning the service on the Virtual Execution Environment(VEE) and monitoring the deployed services, adjusting their capacity through manipulation of the number of VEE instances and their resources based on compliance with the SLA.

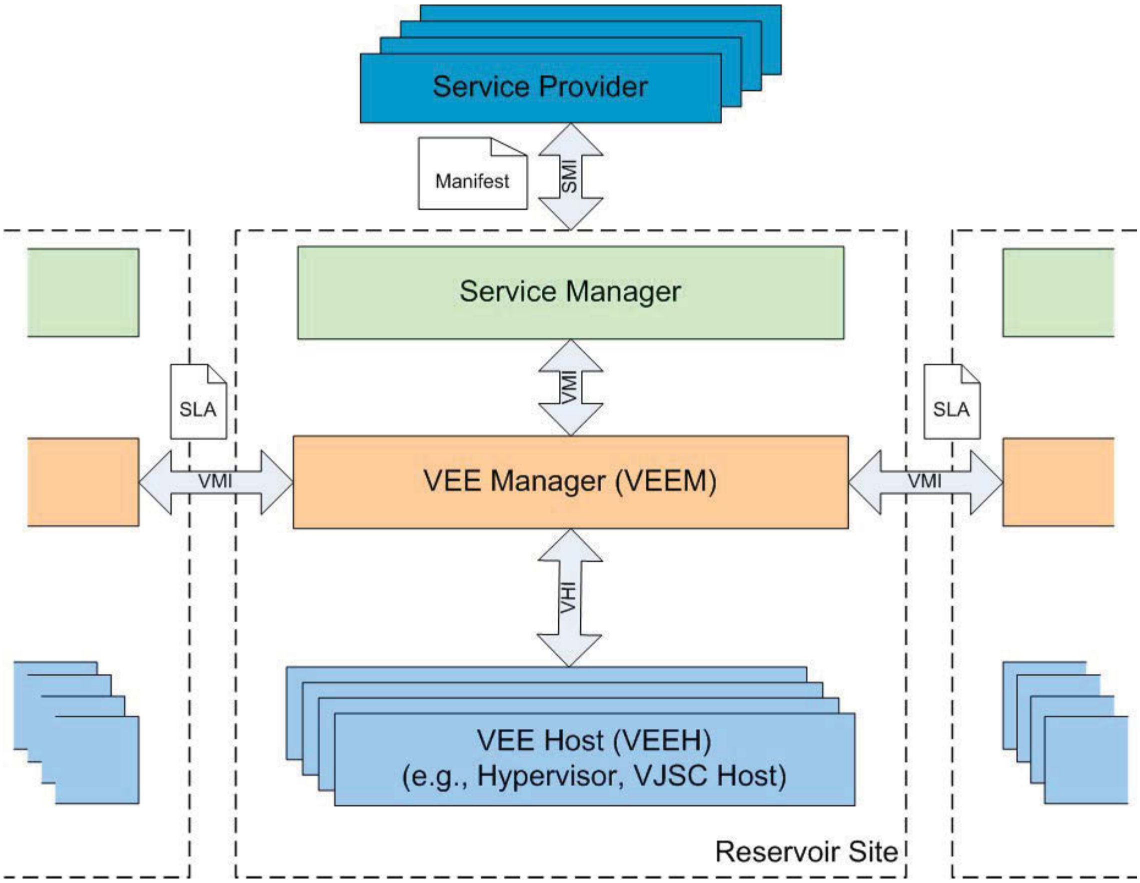


Figure 2.4: Reservoir Architectural layers. (Rochwerger et al., 2009)

At a lower level of abstraction to the Service Manager is the Virtual Execution Environment Managers (VEEM) which is responsible for the optimal resource placement of the Virtual Execution Environments (VEE)'s in the Hosts, subject to the restrictions placed on them by the Service manager. As such interacts with the Service Manager above and the VEE Hosts below. The key concept for the VEEM is that it can place the VEE's on any hosts, even at remote locations. To accomplish this, the VEEM takes on the role of the Service Manager for the remote sites.

At the lowest level of abstraction, the VEE Host is responsible for the basic control VEE's and their resources, including monitoring, the resource allocation, and migration of the VEE's. Each Host covers one particular type of virtualisation technology, and support isolated virtual networks that span various Hosts and different sites.

While this system can manage the failure of providers in the federation and dynamically increase resources for individual applications. The management would come at a cost, as the federated cloud requires that its service managers, VEEM's and VEEH's in the reservoir sites to run continuously across the infrastructure providers, even when not being utilised by the system. Also while there is mention of negotiation between providers and the service managers the actual method negotiation is not described.

2.5.1.2 Cloudbus Inter-Cloud

A second example of inter-cloud federated management, is an approach based on an open market, with brokers that co-ordinate the cloud resources for users through Service Level Agreements. The Cloudbus project(Buyya et al., 2009) involves multiple tools for Cloud management, including a platform as a service toolkit (Calheiros et al., 2012) for developing and deploying cloud computing applications, a simulation tool (Calheiros et al., 2011), a workflow engine, a market maker/meta-broker (Garg et al., 2008), and a storage system (Broberg et al., 2009). For federation Cloudbus also proposes an architecture for the interaction with multiple cloud providers, through a market orientated cloud exchange, that would allow services to be negotiated through SLA's to increase the scalability and performance of provisioned services (Buyya et al., 2010). The architecture of the Cloudbus inter-cloud includes *Cloud Coordinators* (CC), *Cloud Brokers* (CB), and the *Cloud Exchange* (CEx).

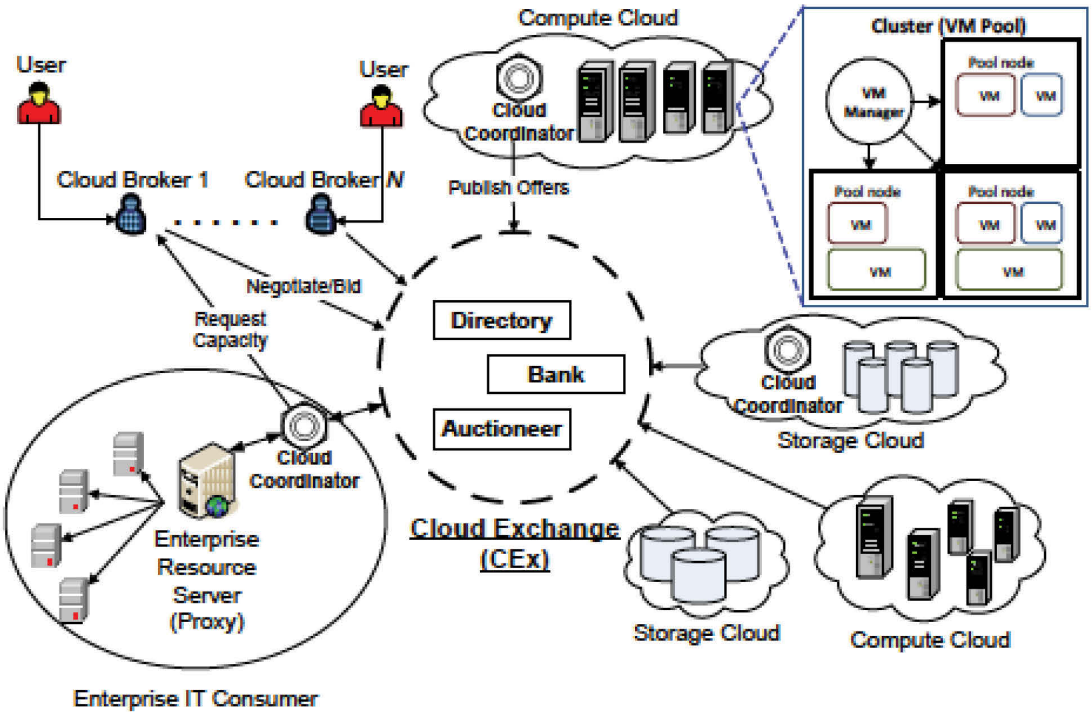


Figure 2.5: Inter Cloud architecture in a federated cloud.(Buyya et al., 2009)

The cloud coordinators are responsible for the management of the resources in the federation, as well as a deployment and management environment for the applications in the federated clouds. This includes the scheduling and allocation of resources for application contracts, with the attendant performance monitoring through a sensor module and a discovery and monitoring module. The scheduling and allocation module interacts with the application composition engine, which allows developers to create and deploy applications in the federated cloud. Finally there is a market and policy engine which stores the service terms and conditions that are supported by each cloud in the federation, which supports the pricing, accounting and billing modules to determine pricing, account the usage and charge the users, respectively.

The Cloud Broker entity of the architecture acts on the behalf of users to discover suitable service providers through the cloud exchange, to negotiate with the cloud coordinators for resource allocation that meets the user QoS requirements. This is accomplished through three modules: the user interface, the core services, and the execution interface. The user interface, involves the application interpreter which translates execution requirements, the task inputs, information about task outputs, and the desired QoS, the service interpreter. This interprets the service requirements such as location and service type, and the credential interpreter, which is responsible for security. The core services, which enables the main functionality of the broker. The modules responsible are the Service Negotiator, the Scheduler and the Service Monitor. The service negotiator is responsible for negotiating the services from the cloud exchange. The scheduler determines the most appropriate cloud services for the user. The service monitor manages the availability of cloud services for the user, discovering new services when available.

Finally there is the Cloud Exchange (CEx), which is the market maker for the cloud brokers and coordinators, evaluating broker demands against coordinator supply availability. Negotiation in the cloud exchange supports various auctions and commodity markets based on SLA's, which is described in greater detail in Section 2.6. The SLA's specify the services in terms of agreed upon metrics, incentives and penalties for meeting or violating specifications. The cloud exchange also provides a banking system to ensure secure financial transactions.

The Cloudbus Inter-Cloud architecture also attempts to provide some risk management via computational risk management through violation penalties attached to the SLA's. In Cloudbus, the broker is responsible for selecting optimal resources and deploying and monitoring job execution on selected resources.

These two approaches to federation are by no means the only attempts. Other attempts include both CLEVER and OPTIMIS, which are described briefly below.

2.5.1.3 CLEVER

CLEVER (CLoud-Enabled Virtual EnviRonment) is an approach that has grown out of attempts to simplify the management of the virtual infrastructure of private or hybrid clouds while still providing interfaces for the integration of external clouds (Tusa et al., 2010). The architecture is similar to the reservoir entities, with Host Managers(HM) being responsible for the deployment of Virtual Environments (VEs) as well as the migration and monitoring of these VEs, while the Cluster Manager(CM) provides the interface between the HMs and the clients, while providing overall monitoring of the cluster. Communication between the CM and the HM utilises the Extensible Messaging and Presence Protocol (XMPP). In a horizontal Federation scenario, the use of XMPP would allow CMs from one CLEVER based cloud environment to control the HMs of another CLEVER cloud based environment (Tusa et al., 2011). Again like the reservoir project, the CLEVER federation relies on the CLEVER middleware to be deployed on any cloud that would be involved in the federation, and would also demand a degree of trust between the member clouds.

2.5.1.4 OPTIMIS

Much more recently the OPTIMIS (Ferrer et al., 2012) toolkit for cloud services has been proposed. While the Optimis toolkit maintains a similarity with the Cloudbus Aneka toolkit, and its focus is on service creation and deployment. The proposed research also includes the evaluation of risk in the context of contracting with new infrastructure or service providers. This concept of risk evaluation, based on an historical performance reliability judgement is a confirmation of the direction of this thesis as to the viability of the risk management proposed in the later chapters.

At the same time the management scope of all of these projects is limited to the cloud providers themselves, and does not consider the necessity of guaranteeing the service connection between the cloud providers and coordinating the bandwidth interaction.

2.6 Market proposals in Management

There has been a fair amount of research done on auctions and negotiation mechanisms for service level agreements with service providers, particularly in the grid environment which was the precursor to the cloud environment. There are currently multiple types of auctions available for these interactions and agreements for service provisioning, from single item auctions such as English, Vickery, Double and Combinatorial auctions, which focus on more complex valuations across multiple resources, and adapting and ensuring QoS requirements in cloud computing SLA's (Stantchev and Schrpfer, 2009). Initial research into English, commonly known as first price auctions, Vickery, known as second price auctions by Grosu, et al. (2006), shows that while English auctions favour resources, and Vickery favours users, the Double auction benefits both users and resources. Further research by Kant and Grosu (2005) has compared three types of double auctions, the Preston-McAfee Double Auction, the Threshold Price Double Auction, and the Continuous Double Auction. The research determined that, in terms of performance and economic efficiency, that the Continuous Double auction was an improvement for users and providers in terms of utilisation, profit and user budget.

There are various approaches to creating an open-market for Grids based on the Continuous Double Auction, such as SORMA (Nimis et al., 2008), Cloudbus (Buyya et al., 2010; Garg et al., 2008), GridWay (Rubio-Montero et al., 2007) using different strategies such as dynamic pricing based on historical usage (Pourebrahimi et al., 2008), Zero-intelligence plus and Q-Strategy (Borissov and Wirstrm, 2008).

In telecommunications, market proposals have involved double auctions (Gibney and

Jennings, 1998), but tend to favour second price auctions (Maille and Tuffin, 2006; Semret et al., 2000). These network based markets are focused on bandwidth allocation, in mobile wireless networks (Ming et al., 2009), or optical networks (Hedayati et al., 2011).

Currently there is no real consensus on the most appropriate strategy for the market, and the focus in both the grid market and the telecommunication market is on much lower level resources, such as bandwidth and processing. In terms of a complex service, which can involve multiple unique service requirements, the use of auctions to determine price has the potential to become difficult. The uniqueness of the service contract precludes the ability of the auctioneer to accurately rank the *bids* and *asks* in an auction.

To address the issue of uniqueness with regards to non-functional properties in clouds, some researchers have been working with *bargaining* negotiation strategies(Venugopal et al., 2008). Bargaining negotiation strategies involve both offer and counter offer from both the user and the provider, and allow for the consideration of multiple targets in negotiation. A recent paper confirms the concepts that drive this thesis, in that Dastjerdi and Buyya argue that for negotiation in cloud resources, a judgement of reliability is also a necessary target (Dastjerdi and Buyya, 2012). However, for the provisioning of dynamic services negotiation through bargaining is potentially expensive in both terms of negotiation time and communication traffic as offers.

However, pricing strategy is only one part of the issues surrounding the implementation of a global open market. Further issues include a common structure for the negotiated SLA's language, and mapping between the different layers: from Infrastructure to Service provider to user.

2.7 Current Commercial Network and Cloud Resource Management Systems

The following section is an attempt to describe the current state of in practice management of telecommunication networks and cloud computing environments. As such it draws more from the authors previous experience in the telecommunication field than research.

Current network management systems are designed from the perspective of managing a single network or a group of networks owned by a single company. Modern telecommunication services however, are global, involving more than one telecommunication company and/or service providers. These services require monitoring and control across the entire service to ensure quality of service. This leads to a Inter-domain problem, as service providers network management systems cannot manage the parts of the service that is not in its control.

For large telecommunication providers, the problem is exacerbated by the issue that the networks function and purpose is divided by the company defined domains, such as residential, wholesale, core, small to medium business and business. This often results in the management of the network being the responsibility of multiple management groups increasing the problems of management. This is an extension of the inter-domain problem just highlighted, with the additional concern, that the common solution of the inter-domain problem, centralisation and integration of management systems, is not feasible, for two reasons. One, that the domain owners place security and administrative barriers to the integration, and two, centralisation of so many disparate network management systems would be completely impractical, requiring an unheard of integration effort and subsequent unacceptable loss of control for the individual domain owners. In the authors experience the solution often used is the negotiation of contracts and standards which set the terms and parameters of all future instances of a particular service. The negotiation of which is typically offline and generally includes proposal, review and approval phases for every domain with the attendant delays to allow the various stakeholders to include their input. These delays, which in the authors experience are in the time frame of days are when dealing with an entirely new service such time can be seen as worthwhile. The problem that the author experienced is that this methodology is applied to the building of specific instances and changes to a service, so that any difference in service fulfilment requires the same process, regardless of the networks technical capabilities.

Network management systems in industry are focused on discovering the root cause of faults. The root cause being the network or service problem which causes a cascade of faults in subsequent devices (Steinder and Sethi, 2004). Current network management systems utilise various methods in collating and analysing this data, from simple rule matching, information models(Strassner, 2003), and ontologies(Green, 2004). However this analysis of root causes depends on the data being received from the network. If the problems being reported are fail-over effects from another domain, then any analysis of the root cause either traces the cause to an external source, or as the view of this network management system is on its network alone, can report performance problems in a perfectly healthy device.

Reporting and analysis is done after a problem is detected this is considered to be reactive, as such current management systems are reactive(Wallin and Leijon, 2009). By this the author means that network management systems respond to changes in the network management system based on performance triggers. Typically network management systems follow a similar pattern, which is report the problem or alarm, collate all alarms that might be relevant, discover the root cause through analysis, and then report the discovery for evaluation and manual intervention. In more advanced systems the network management system can reconfigure devices and services, from either static templates (Kim et al., 2011), or policy models. In the case of static rule matching templates, this automated reconfiguration can lead to further instability if the analysis done by the network management system is incorrect, as the pre-prepared solution will also be incorrect. However, with certain defined parameters such static templating can be effective. Such examples of pre-programmed responses include the networks designed redundancy, with the re-routing generally being automated at the router level. For Telecommunication and Cloud computing companies the pre-programmed responses generally also include new provisioning or the assigning of some template resources, such as bandwidth for telecommunication companies, or number of virtual machines for a cloud provider, to a particular client.

For network providers examples of the management tools that utilise these reactive management techniques include: Nagios (Josephsen, 2007), and the HP suite of network management tools such as NMi, which derive from HP Openview. Both management systems utilise a centralist view of network management however Nagios being an example of open source based network management that relies on individual configuration to facilitate the fault reporting and again relies on individual configuration for the implementation of limited template based analysis. The network management based on HP Openview however, has from the late 90's tried to provide some analysis and fault management based on event correlation (Sheers, 1996). This has been expanded to provide some limited control through network configuration changes and the ability to specify policies (Hewlett-Packard Development Company, 2009). It should be noted however that the policies are mostly directed to network integrity checks and do not integrate with the allowed configuration changes.

For cloud providers, the management is in a better state, as the problems for configuration and provisioning are extensions of IT based management that has continued for decades. The introduction of virtual machine technology has been accompanied by configuration and performance management software as well as a new paradigm where services are dynamic and are purchased for a finite time or for a specific task(Armbrust et al., 2010). Solutions for common problems, such as load balancing, are managed in very similar ways to current physical server technologies. The largest task management concern of public and private cloud environments is scheduling. As a result of this dynamic nature of services, the public cloud environment is already moving towards a market based environment, with Amazon introducing spot pricing (Amazon, 2012b) which is an auction type purchasing structure. The basis of spot pricing is that the user sets a maximum price that they are willing to pay for a resource, based on the current advertised price and known history, and the resource will run until the job is finished or the price rises to greater than the users maximum. For public clouds to be practical there has always been a need for performance and configuration management to be utilised by the cloud users. Amazon Elastic Computing 2 (EC2) for example, provides an API enabled Amazon Web Services (AWS management console, which allows the user to configure the virtual server, monitor tasks on the virtual server, and receive performance information. For hybrid clouds, there are various portal interfaces which leverage the public cloud management APIs to provide integrated management with the users legacy infrastructure. HP is commercially releasing monsoon(Yan et al., 2011), which incorporates policy driven management and maintains a pool of cloud infrastructures, managing the different proprietary interfaces of different Infrastructure as a Service (IaaS) providers.

It is these proprietary interfaces and the lack of interoperability that is a continuing problem for responsible inter-cloud management. While individual clouds, such as Amazon

EC2, Microsoft Azure(Microsoft, 2012), and Google app engine (Google, 2012) already allow replication, and on demand scalability, in their applications and virtual machines as well as the internal migration through availability zones (Amazon, 2012a). The ability to seamlessly move virtual machines to competing clouds is currently stymied.[1]

However, researchers are already approaching the technical issues in managing the technical interaction involved in the migration across domain boundaries, providing routing as a service(Chowdhury and Boutaba, 2010) and of migration virtual machines over wide area networks without losing service(Hao et al., 2009). On a more basic level, the Amazon cloud and current research toolkits such as Cloudbus Aneka(Calheiros et al., 2012) accommodate multiple current virtual machine architectures (such as VMware and Xen).

2.8 Technologies for implementing management systems

2.8.1 Agents in Telecommunications

The inherent distribution of a telecommunications network, and the multiple services that they provide, corresponds well with a multi-agent systems

> ability to cooperate towards multiple goals (Wooldridge, 2009).

In reality the current management of telecommunications networks is done through agent based technologies. From still in practice simple reactive software agents, employed by SNMP to monitor particular devices or servers from the early 90's, to the more advanced research concepts of autonomic management, realised through autonomous agents described as the Autonomic Management Element (AME). While the previously mentioned policy based management describes the ability to configure network or cloud devices from policies derived from business rules and does not directly concern itself with the implementation methodology of the configuration. It's adoption by autonomic management means it is effectively implemented through the same AME. Further, various agent architectures have previously been put forward for service and network management (Turner et al., 2010; Giri et al., 2009; Lee et al., 2008).

2.8.2 Agents in Cloud Computing

All of the previously mentioned systems for service management in the inter-cloud, such as the Reservoir project, Cloudbus, and SORMA, utilise agents. The agents are used implicitly in the case of reservoir, with individual Service Manager agents negotiating towards a Service Manifest. While cloud broker agents interact with the cloud co-ordinator via auctions in the case of Cloudbus, and bidders and sellers agents are utilised in the case of SORMA.

---

[1] After the time of submission, Amazon released the ability to move between their geographical regions, by manually downloading the VM and uploading in the different cloud. However this is still an entirely manual and offline procedure. The ability to migrate live VMs is still unsupported.

SERA(Ejarque et al., 2010), is one system that explicitly utilises a multi-agent system in the integration of customer jobs and resource management. The distributed SERA architecture instantiates both Job agents, whose responsibilities include getting resources, scheduling, stopping, suspending and running jobs. The Resource agents, whose responsibilities include monitoring scheduled and running jobs, registering resources and limited recovery. This limited recovery is defined as triggering appropriate policies through the agents rule engine, with policies that must be defined statically at design time.

Overall there is almost a universal adoption of agents when dealing with marketplace requirements. As the focus of this thesis is the management of end-to-end performance of services in a global market, it is natural to utilise agent technology for the design of a system capable of satisfying the proposed goals of this thesis.

2.9 Summary

The research presented in this chapter was concerned with the development of management strategies in the context of increasingly complex telecommunication services. As the aim of this thesis is to provide overall management coverage, through responsibility, over the wide range of available telecommunication and cloud services, this chapter has attempted to cover approaches in traditional network management as well as the cloud computing environment.

Initially this chapter attempted to demonstrate the development of *management* research, in the context of telecommunications and cloud services, as an area in which the focus has changed over the years. Initially focused on the optimisation of networks as a whole, research in the field of management has slowly shifted to focus on individual services. Early attempts at managing the service performance across multiple domains were unfortunately undermined by immature technology and organisations that were slow to embrace the advantages. Concurrently, as the technology improved and as organisations realised the value of telecommunications to their business, researchers increasingly focused on integrating the management of networks with the goals of the businesses and organisations. The result was Policy Based Management, which introduced the ability to semantically coordinate business goals with the configuration of devices for the provisioning of services in telecommunication networks.

However, as services became increasingly complex and networks grew progressively larger, policy based management was considered insufficient on its own. While PBM increased the organisations capability for service management, it was limited by a centralised approach, derived from the traditional view of whole network optimisation, that proved inefficient in responding in a timely manner to network and service problems. Research moved towards decentralised management. In particular, Autonomic management proposed the concept that management occurs through autonomous agents that *self* manage the devices or network section they are responsible for, under policy guidance and cooperation from other agents. The combination of policy based management and autonomic management shows great promise in the management of individual networks, however the coordination required for balanced policies for global services consisting of autonomous networks introduces further complexity that current autonomic management is yet to address.

One proposed approach is Federation, the building of service level agreements between autonomous management systems to coordinate the management of services across the federation members. However, in autonomic management federation is in its infancy, with issues identified with semantic equivalency, federation authority, trust and the degree of management that would be allowed to federation members.

In the realm of Cloud services, federation has been proposed as a means of dealing with the perceived eventual scalability problem with services in an individual cloud. In the cloud environment, researchers have developed management agents, strategies and toolkits for the coordination of services across multiple cloud providers. In this manner they attempt to avoid the semantic problems involved with autonomic federation. However, this particular view of federation is limiting, as the focus is solely on cloud services and ignores the other, just as vital, resources (such as connectivity) that comprise modern services.

Inter-cloud federation has also proposed market based negotiation strategies for service contracts. However, while current approaches based on auctions and bargaining are appropriate to negotiation for the well defined services that are available in a cloud environment, they are inefficient when dealing with the variabilities of current services. At the same time, auctions and bargaining raise concerns of timeliness when purchasing on demand services.

The industry has been slow to adopt the proposals that have been presented in this review. Indeed the greater percentage of management systems in practice are reactive systems that involve constant human interaction. As technology continually improves, individual management systems have evolved to greater automation in certain areas, such as the cloud environment, however overall it is believed by the author that the management gap between practice and research is widening. While the exact reasons for this would be a thesis on its own, the author believes that a primary reason is incentive or rather the lack of it for individual providers.

For the management of services that cross multiple domains, it is believed that an alternative approach is needed, one that focuses entirely on the performance of the service unconstrained by the business concerns of the individual organisations and yet able to incentivise them. As opposed to autonomic management and the autonomic federation, the coordinating system should limit their responsibilities to the performance concerns of the service. Rather than controlling the whole service across the federation as in the inter-cloud approaches, the system would need to work through negotiation across a market with the management systems of the individual providers, structuring interaction to certain basic requirements. By advocating total coverage but limited control, the system would require the ability to judge the risk in negotiating with the various individual providers. This risk management would serve two purposes, improving the reliability of multi-domain services from the perspective of the user, and providing the needed incentive to individual providers for a reliable service.

Chapter 3

System

The research presented in the previous chapter provided an overview of the advances in service management in both telecommunications and Cloud computing. However, the variety of purpose and abilities in the management systems showed a deficiency in addressing the requirements of multi-domain services of composite types, specifically in terms of service responsibility. This thesis proposes the *Complete Autonomous Responsible Management Agent* (CARMA) framework to address this deficiency.

Initially, this chapter attempts to define the requirements and entities of a system framework responsible for the negotiation and management of modern complex inter-domain services. In the context of negotiation this chapter focuses on the options and constraints of contract inception and execution in a multi-domain environment.

Further, the marketplace requirements are defined and the alternative approaches in literature are re-examined with the result that a market framework is proposed.

The architecture of the CARMA framework is then introduced and the roles and responsibilities of the entities are discussed along with the alternatives. The architecture of the framework is developed based on the research presented in chapter 2, their underlying assumptions and industry knowledge.

Finally, the underlying assumptions and risks of the CARMA framework are discussed.

3.1 Requirements for a Market based Management System

The goal of the system is to provide responsibility to multi-domain telecommunications. From this goal and previous analyses (Mearns et al., 2010b; Sheridan-Smith, 2007) into the architectural requirements of network management systems, the following requirements of the architecture have emerged.

- The Management must be automated.

- The Management will need to contend with increasingly complex collections, or bundles, of interrelated services and bridge the gap between service management and network management.

These properties lead to the determination that the management architecture must be inherently decentralised. Together with the marketplace requirement, this has suggested that a possible solution to a Multi-domain management system is one which relies on the use of independent agents. Architectural requirement goals determined to be fundamental for a successful architecture of a management system include:

- Responsibility, which is defined by two sub goals, that of:
  - Risk management, The ability to judge the inherent risk in contracting and managing services across domains. This risk is seen as a combination of all the factors that can influence the service, from the service providers performance, current and historical, the quality requirements of the service and the interaction of the various providers involved in the service.
  - Resilience, Which is the ability of the service to recover from failure. This is considered to be above the single service providers focus on the reliability of the service which also covers the providers service redundancy. However, Resilience for the inter-domain is focused on the point at which the reliability of one provider cannot be ensured and proactive steps need to be taken to recover the service and maintain the minimum required quality for the overall bundle.
- Scalability, Any system that has to operate in an environment of diverse networks in a global environment would have to contend with issues of scalability. Further, as bundles become increasingly complex the number of involved providers would likewise grow. This complexity would add further burdens on the scalability of any management system.
- Efficiency, In this context efficiency is seen as the ability to manage or affect the utilisation of the disparate resources involved in the bundled collection of complex services. The term efficiency is used to differentiate from the goal of greater utilisation of all providers, as the minimum requirement of this system is to increase utilisation for the more reliable providers, as an incentive.

First of all this section will identify the entities that compose the inter-domain environment and present the requirements that the market based management system needs to fulfil. The requirements cover the fields of market specification, provider interaction through contract specification and the acceptance of responsibility.

3.2 Entities and their Relationships

This CARMA framework is built on the work done by Les Green (2004) on a marketplace negotiation framework of service level agreements for single services. The focus of this thesis extended the work into the realm of composite services, which is the extension of the single service idea that aggregates multiple services and focuses on the interaction that occurs at the Inter-domain level, through the use of an agent based market. Central to this argument is the concept of a dynamic *Bundled Service*, which is defined as a complex service that involves multiple providers. In order to manage the interactions between multiple providers for a bundled service in a market, a service aggregator or broker is needed. To follow from the concept of a bundled service, the name for this broker has been chosen as the *Bundled Service Provider*. Continuing this argument it follows that the entities that represent the providers of the single services that comprise a bundle are labelled *Single Service Providers*. However in the telecommunication environment, it is known that the resources that the single service depend upon might not be entirely owned by the single service provider. Consider the case of a global video conferencing service, the conferencing processing software is likely to be situated, relatively independently, in data centres around the world, to minimise the effect of network lag affecting the service. It is highly likely, in this scenario that the video conferencing provider has a contract with the data centre owner, with regards to bandwidth and server redundancy. These service resource providers are entities that own resources that the service providers are dependant upon.

The focus on market directed negotiation is predicated by the QDINE system in Les Green's (2004) work. In the current environment, telecommunication service contracts are negotiated individually between the provider and user. Traditionally this has been an offline process as described in Section 2.7 but has become more automated as providers take advantage of Internet based provisioning systems. The continued increase in the number of Cloud Computing and telecommunication providers, has seen competition arise on a global scale, with this competition beginning to drive pricing. An example of this is that the IaaS Cloud Environment by Amazon (EC2) started with flat pricing for their services, then moved to pricing based on service differentiation, and has more recently, introduced an auction based pricing scheme which is described in greater detail in the literature review Section 2.7. At this stage it is not difficult to envision that this continuing trend will result in global markets for telecommunication and cloud services.

The CARMA framework proposes the use of intelligent agents to fulfil the risk, scalability and efficiency requirements stated by Section 3.1. Intelligent agents in a multi-agent system embody properties that are considered useful in meeting the requirements.

1. Agents are automated, requiring no human interaction to function.
2. Agents are both reactive and proactive, meaning that they can both respond to changes they perceive in their environment and they are able to exhibit goal-directed behaviour in order to fulfil their design objectives.

3. Agents can exhibit autonomy in their goal directed actions and it is these independent actions, that can encourage competition in the marketplace and manage the resilience of the contracted service.
4. Agents are an inherently distributed technology, which implies a greater scalability than that provided by a centralised system.

Figure 3.1 shows the entity relationship for the principal entities of the system. Each of the principal entities have associated agents. The agents are described in more detail in Section 3.5.

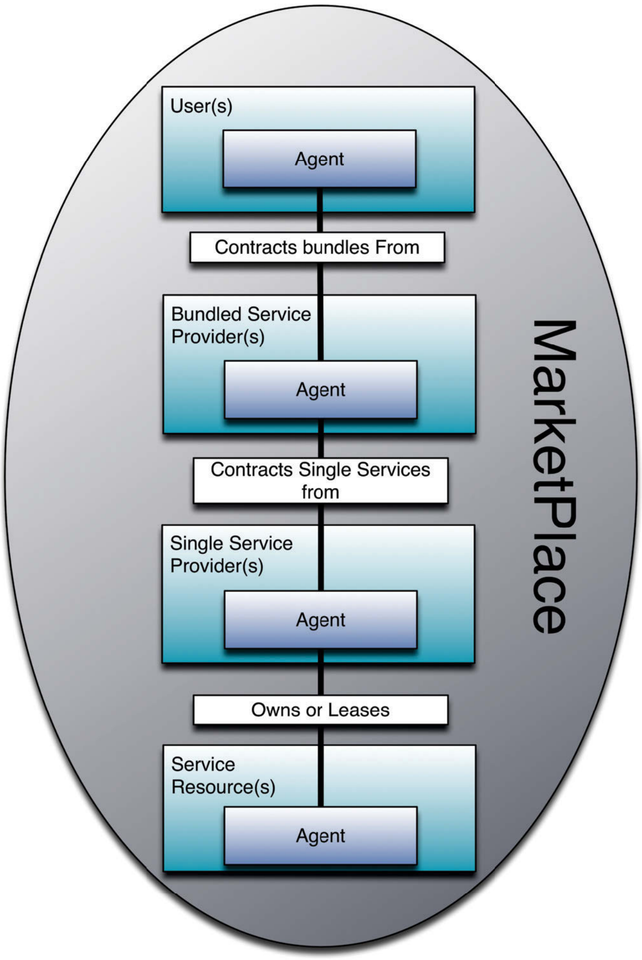


Figure 3.1: Entity Relationship Diagram of the main Entities in CARMA.

Below is a brief overview of the required functionality of the entities in the system.

- **Users:** make bundled requests via their user agents.
- **User Agent:** (UA) negotiate with bundled service providers to obtain services at the required quality and price.
- **Bundled Service (BS):** as described by the TeleManagement forum (TMForum, 2010), and extended to include cloud services, is a collection of multiple services offered with incentives, that include extra organisational issues.

- **Bundled Service Provider (BSP):** a service provider which provides bundled services to users, and negotiates with (a number of) single service providers including application and connectivity to provide the user's service, at the required quality.
- **Bundled Service Agent (BSA):** is a an agent assigned to a User request, and to the subsequent management of the bundled service.
- **Single Service (SS):** The contracted SLA with a single service provider. For the purpose of this project we define the service primitives to be; Remote Application (RA), Remote Virtual Machine (RM), Remote Virtual Storage (RS), Video Conference (VC), Audio Conference (AC), and Network Connectivity (NC). Examples in (Green, 2004).
- **Single Service Provider (SSP):** is a provider of single services. They may have many single services available.
- **Single Service Agent (SSA):** An agent in control of the single service for the selected service time period.
- **Service Service Resource Provider(SSR):** Maps to a resource of the primitive single service type. Note: Within each service type primitive operational decisions are made with regards to response times, bandwidth, and processing capabilities. This variability will give rise to varying QoS, which can only be judged by measured performance.
- **Single Service Resource Agent (SRA):** An agent for a single resource. The agent keeps track of the resources allocated for the single services at the particular service times.

The structure of this system is hierarchical, with each entity only interacting with the entities above or below them. As such there is a need for differing specifications of the bundled service contract and different types of market interaction between the entities.

3.2.1 CARMA Entity Goals

The goal of CARMA is to create an environment that allows another level of awareness and control in a contracted complex service. One that monitors and controls the interaction of the various providers that comprise the complex or bundled service. There are various options for interaction between the entities that have been identified in this environment. For example, the single service providers could interact with each other directly, with the responsibility for the service devolving onto the single service provider that initiates the service request. This interaction could happen with or without a marketplace. However, with the dynamic nature of services being discussed, without a marketplace the access to providers would be limited to pre-existing contracts negotiated off-line. This results in poor competition and pricing for the user. Also, due to the nature of the Inter-domain problem, each of the single service providers and brokers will have different goals. For example, the single service provider wants to achieve the greatest utilisation for its services that it can, while a Bundled Service Provider is looking to get the highest quality service for the lowest price. In this environment the federated approaches, previously discussed in Section 2.4.5, where the single service providers either cooperate to provide the service or an external provider controls the services in the federation can become complicated. In a federation based on cooperation, the conflicting goals leads to increased complexity. In the approach based on a separate authority, while appropriate for services of a single type, the sheer cost of reserving all of the different types of providers in a modern telecommunication service makes the solution impractical for bundled services of different types.

3.3 Contract Specification

In the requirements section 3.1, the focus has been on the entities of the system and their goals. Equally important is the entities ability to interact with each other. As the structure of the system is layered, entities only interact with entities that are above and below them. A formalisation of this concept can be seen in Strassner's Policy Continuum (Strassner, 2003, p.367), which exposes 5 layers of policy from the high level Business view to the low level Instance view and is described in greater detail in Section 2.2.

For this system the particular levels of interaction between providers can be thought of as *views*. The interaction between the user and the Bundled Service Provider is the *user view*, while the interaction between the Bundled Service Provider and the Single Service Provider is the *provider view*, and the interaction between the Single Service Provider and the Single Service Resource Provider is the *resource view*. For each of these views, different levels of specification is needed to define the terms of a service level agreement. With the user view being the most abstract, and the resource view being the most concrete. Abstract in this context is used to describe the fact that at the user view, there are certain specification requirements that are not necessary for users or business owners to know in order to negotiate a service.

Taking the example of a Virtual Private Network (VPN) service, the user of the service has no need to specify the protocols that are chosen for their use. At the provider view, which is a more concrete specification, the types of protocols being used, such as MPLS need to be defined. At the resource view, analogous to the policy-continuums system view, device and technology independent but specific connectivity information, such as termination points needs to be defined. While the differing views contain concepts that are common between the views such as the concept of a service, there currently no semantic equivalence between the layers of abstraction. This leads to a further requirement of contract specification, which is a need to properly adapt and mediate between the views (Strassner, 2003, p.112).

The adaptation and mediation between the user view and the provider view utilises both a common information model and graph transformation to provide contract semantic equivalence and is discussed in greater detail in the design Section 4.3.2.

As well as the general requirements for interaction between the various viewpoints of the entities, the second part of the contract specification is a definition of the scope of the services that can be provisioned by CARMA. In order to cover the scope of the service level agreements some understanding of the negotiation issues surrounding the provisioning of bundled services is required, the basis of which is an understanding of a bundled service. The Tele-management forum, [2] (Forum, 2012) defines a bundle as a:

> collection of services with added incentives, that may include extra organisational issues.(TMForum, 2010)

Breaking this definition down shows that first there needs to be some definition of a single service in this context. At the user view, there needs to be some common parameters for specification regardless of service type.

While the varied nature of services in a multi-domain environment means that individual services have a multitude of specialised parameters, in all services in a telecommunication environment there are certain parameters that are universal. For example where the service is located, be it geographical, or IP based. Another basic requirement is what kind of service is needed, which would form the basis for all further specialisations. In the context of dynamic services, the length of the service is also important. Finally in context of guaranteed performance, some performance indicator would also be required.

For this specification the generic parameters are defined as:

- Path (Locations: from, to),
- Type,
- Quality,
- Time.

For Type, in the context of the user view there has to be some generalisation of services that are available. It is envisaged that the user view level terminology is designed for users that are not natively familiar with technology. At this level there are six defined types of service:

- Conference (audio and video), refers to any video or audio conference service that is controlled remotely.
- Application, refers to any piece of software that is installed remotely and runs independently from the users system.
- Machine, refers to any remote bare bones machine running on real or virtual processing capacity.
- Connectivity, refers to network bandwidth services.

---

[2] which is a standards body dedicated to improving service provider operations, through the introduction of a management framework and common information model

- Storage, refers to any remote storage service.

In the context of service delivery few of the service types are ‘atomic’, rather they are themselves composite (if not bundled) services, with almost every type of service listed above requiring another service to run. For example, a remote machine service, such as a virtual machine in the cloud requires the connectivity to be assured between the user and the cloud.

The Path and Time parameters simply refer to the location of the service and its duration. At this level it is expected that the user specifies either one or two general geographic locations. The Quality parameter refers to the services required performance level and, consequently, affect the cost, risk and successful completion of the service.

Referring back to the Tele-Management Forum’s definition of a bundled service, the second part of the definition highlights the possibility for extra organisational issues. For telecommunication based bundled services one of the most important organisational issue is the issue of service dependency. The dependencies or constraints between the services are the basis upon which the Bundled Service Provider judges risk, by exposing the structure of the complex interactions when dealing with individual service failure. The two most common dependencies are those based on time and cost. CARMA defines two more, the reserved services dependencies which defines those services that may be required by a Bundled Service Provider(BSP) to provide the required level of performance quality and the Minimum Quality dependency, that specifies the minimum allowable quality level before which the service is switched to another provider or is allowed to fail. In this thesis the quality levels have been abstracted into the generalities of high, medium, low and best effort. These generalities refer to different performance criteria for different services. For clarity the dependencies are defined here:

Time Dependency The dependencies for a bundled service are most commonly based on time.

Cost Dependency Cost dependencies are also important. This cost will be divided if the defined service is a composite service, or just passed through if the service is atomic.

Reserved Services With regards to required high quality services, run on high risk providers the reserved services operator specifies up to n arbitrary connections, which may, optionally, be required during the bundled service to provide the required level of quality.

Minimum Quality Finally there is the minimum quality level operator. This is the level beyond which service recovery becomes necessary.

The requirements of the contract specification, the various levels of abstraction, and the scope of the various dynamic services that can comprise a bundle, introduces issues in service negotiation that becomes impractical when utilising traditional subscription based service contracts. Following the research described in Section 2.6 and the goals of the CARMA system, the applicability of a marketplace alternative is explored.

3.4 Marketplace Requirements

The entities and relationships previously mentioned in Section 3.1 and shown in the Entity Relationship diagram Figure 3.1, such as the Bundled Service Provider and its agents, the Single Service Provider and its agents and the Single Service Resource Provider and its agents, has to be put into the market context. In the market context, such concepts are applied to formulate service bundle products, define them quantitatively, qualitatively and temporally, price the offers and conclude the transaction. In particular the current state of telecommunication and cloud based services adds further complications to negotiation of services in the area of negotiation between Single Service Providers and Single Service Resource Providers.

As highlighted in Section 2.7 Cloud computing is based around the concept of dynamic services, purchased for a finite time. In telecommunications providers, work on set time leased resources, where the contract for these resources is set via contracts or SLA's that are negotiated for fixed prices or in off-line negotiations. While the period of the requirement times is likely to be quite different, the concept is the same. A fixed price for the resources.

Additionally as mentioned in Section 2.7, the Amazon EC2 cloud environment has moved towards on demand pricing, with its implementation of spot pricing. Spot pricing utilises a bid system to implement on demand prices and it is envisioned, that as the market presence expands, that other Cloud and telecommunication providers would move to on demand pricing, which has the advantage of lowering costs in periods of lower utilisation for the user while attracting more users for the providers in the same time period.

As stated in the literature review Section 2.6 there have been various approaches to marketplace negotiation, from English, Vickery and double auctions to bargaining strategies. However, the analysis of the current state of service offering and the roles and relationships that need to participate in the market, highlight issues with the current auction based approaches. Auctions work best with well defined objects for purchase, which is more complicated in an environment of unique services. Bargaining strategies involve at least two or more steps, that of bids and accept/rejection of the bids. In practical terms this two or more steps both increases the traffic required for negotiation as well as the time.

In conjunction with these issues is the individual requirements of the different layers of abstraction listed in the previous section. While it is envisaged that all three layers of abstraction in contract negotiation, will utilise a market to negotiate services. The requirements of the Bundled Service Provider and the Single Service Provider level, and the Single Service and Resource level would be wholly disparate.

These issues and requirements have led to the development of the market context that includes three sub-markets.

Primary Request For Quotation

Potential bundle users send contracts to a seller agent, who then provides a quotation along with a time limit for settlement. A single user may request quotes from multiple sellers. Similar interaction can take place between Bundle sellers and Single Service Providers (Hall, 2001, p. 105). The primary market lies in the interaction of Users and Bundled Service Providers, and Bundled Service Providers and Single Service Providers. The drive for this market type lies in the requirement that the bundled services that are requested are unique, as different users would have different service needs, that would also likely change over time.

Secondary Clearing House

In this situation a buyer issues a complete service contract which includes the price that the buyer is willing to pay. A seller can accept the offer only after a set delay period (Hall, 2001, p. 143). The secondary market would exist in the same interaction level as the primary market. That being the interaction between Bundled Service Providers and Single Service Providers. The focus of this market is for services that are unused by previously purchased contracts. It is envisaged that there will be situations where services purchased and reserved by the Bundled Service Provider for a particular user, would be cancelled, resulting in surplus resources that the Bundled Service Provider would then be able to reuse or resell to other Bundled Service Providers in the marketplace.

Wholesale Dynamic Posted Prices

This is a seller-driven marketplace where the prices for service access are posted by the providers and, if not accepted within a specified time limit, disappear (Hall, 2001, p. 137). The Wholesale market exists in the interaction of Single Service Providers and Single Service Resource Providers. As mentioned in the above paragraph, telecommunications and cloud providers already exist in a fixed price environment, albeit an offline one, and continuing with this concept while moving it to the market place seems advantageous as it combines the traditional selling strategy with the ability for providers to speculate on global requirements. The idea of a dynamic market is promoted previous work by Green (2004).

Two things are important to mention here: the primary market is implemented as a neutral exchange as there is a requirement to accommodate the wide range of possibilities for bundle requirements (composition, quality, timing, etc.). Secondly, the Clearing house allows additional resource-use optimisation to occur as buyers would be interested in finding ways to leverage performance expectation risk via additional resources, and sellers would be interested in selling capacity that has become available due to operational circumstances.

Lastly, the presence of wholesale market allows for entrepreneurial activity, where Single Service Providers can make a purchase at a discounted price if they have a reasonable belief that a service will be sold as part of the bundle requests they will win in the allotted operating period. This belief can be built using recorded usage statistics and market trends analysis.

However, in order to reap the benefits of the market operation it is necessary to build the supporting architecture, capable of delivering the information and providing control functionality at high levels of performance and scalability.

3.5 Architecture

3.5.1 Development of a Service Architecture

The description of the goals, requirements, contract specification, marketplace requirements and definition, forms the basis for CARMA's architecture. The roles and responsibilities of the entities in the multi-agent based system should conform to the requirements already stated in the above sections. Figure 3.2 shows the overall architecture chosen for bundled services provisioning and management in an open marketplace.

Originally the structure of single service providers and users negotiating strategy follows on from the QDINE work, with users being responsible for negotiating the contracts between themselves and the single service provider.

However, this architecture, and the terminology of the SLA ontology requires that the users have a deep understanding of the technology quality requirements needed. While this is acceptable for users with a high level technical background, such as network operations personnel, it is not conducive to utilisation by the vast majority of service users, for whom the technical knowledge is secondary to their field of expertise. For example an accountant whose expertise is in payroll, and wishes to utilise a cloud based payroll application, should not have to concern themselves with the bandwidth necessary to deliver the data. Therefore an entity that can translate the requirements from user to single service provider is required.

This reasoning added to the requirements of responsibility, and the requirements of independent management, that was the initial driver for the Bundled Service Provider entity. Concurrently, past experience in the telecommunications industry, led to the understanding that not all resources that are utilised in a service is owned by the service provider. In any large company, there are multiple departments that are responsible for different aspects of the companies business, from service delivery to accounting, marketing, and payroll. In most cases the divisions are in relation to the service being performed. In telecommunications, the services that are offered can be very similar and can be differentiated only by the type of customer that the service is for, such as residential or small to medium business users.

At the same time, it is highly unlikely that the telecommunication company will duplicate the resources for the service that is utilised by both departments. Instead common practise is to assign ownership to one department with the other 'managing' the services for which it is responsible. In the cloud computing environment, cloud based applications are arising on platforms that are not owned by the application provider. This has led to the acknowledgement that there is a resource management entity that is separate from the service provider. The following section describes the service architecture development. It defines in detail the roles and responsibilities of the multi-agent system entities and agents.

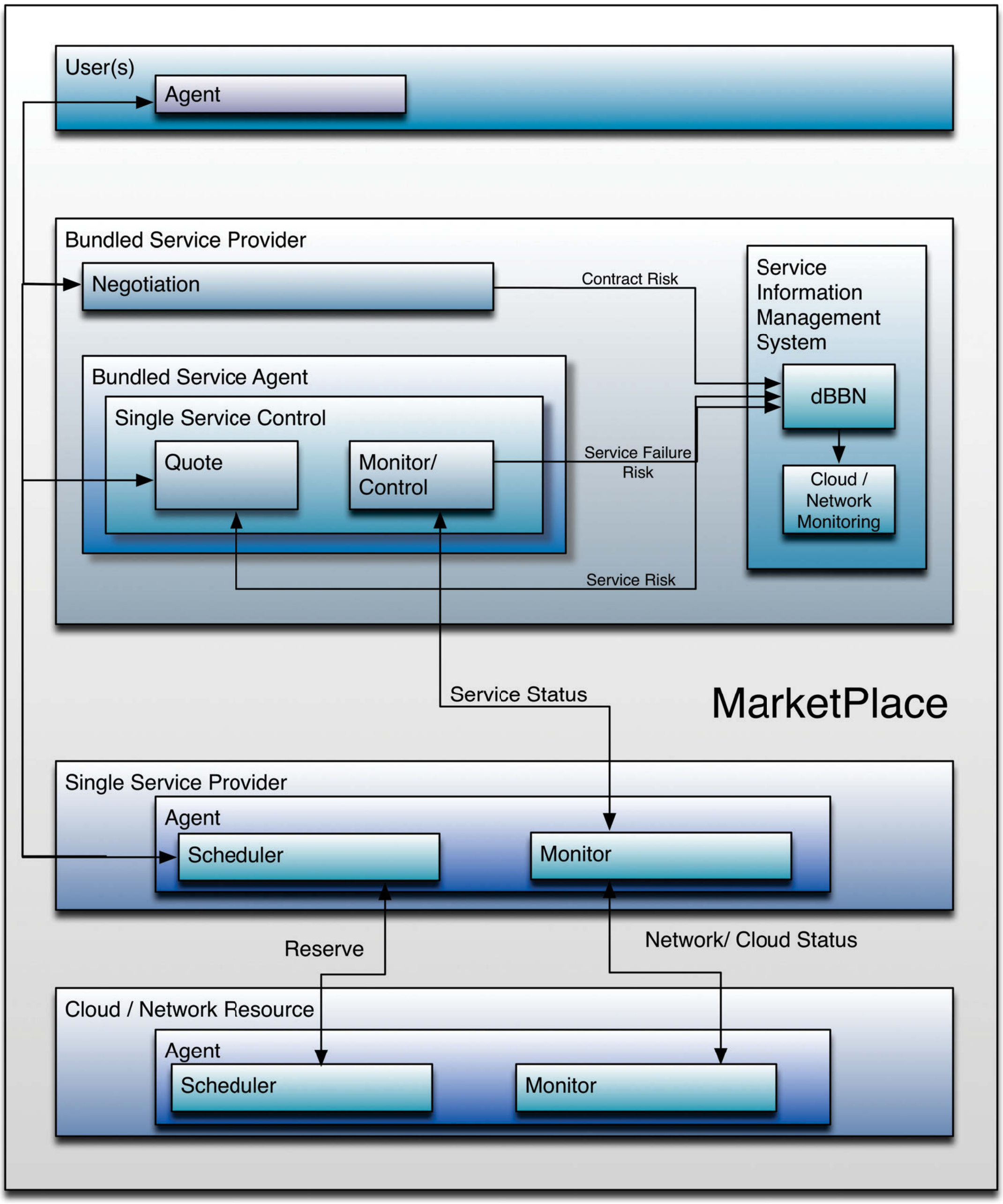


Figure 3.2: Architecture of the CARMA system.

3.5.1.1 Bundled Service Provider

Taken from the requirements for the market based architecture, the Bundled Service Provider concept for telecommunication and cloud computing, is similar to other market based solutions, such as Cloudbus(Buyya et al., 2009) in that the Bundled Service Provider is a *broker* between users and service providers. Unlike traditional brokers however, the Bundled Service Provider takes responsibility for the contracted service during its lifetime. The responsibilities of the Bundled Service Providers (BSP) shown in Figure 3.2, are:

- negotiating the use of the single service providers(SSP),
- judging the SSP's risk, and
- managing the failure of single services in the bundle.

Upon receipt of a new bundle request from the User Agent(UA), the BSP creates a Bundled Service Agent (BSA) to negotiate for and manage each contracted bundled service. Depending on the user's requirements in the service definition, individual single service modules to control the provisioning and management of each single service are added to the BSA's portfolio. The Bundled Service Provider also maintains the Service Information Management System (SIMS) which uses gathered information to judge the risks involved in the bundles.

3.5.1.2 Bundled Service Agent

The Bundled Service Agent(BSA) is responsible for the negotiation and successful completion of one contract. The goal of the bundled service agent is to get the highest quality service for the user at the cheapest price. The BSA uses the SIMS to judge the risk of the various single service providers offering services that match the required services, and requests quotations from the more reliable Single Service Providers (SSP) and present the user with a quote for the bundle. The BSA then monitors the service through its lifetime, through monitoring messages agreed to in the contract and sent from the Single Service Providers.

In order to manage failure from the bundled services point of view, the BSA utilises both renegotiation of, and, the lowering of quality (performance) requirements for the service. Renegotiation swaps providers upon the receipt of negative performance information from the Single Service Providers.

3.5.1.3 Single Service Provider

The goal of the Single Service Provider(SSP) in this environment is to increase the utilisation of the resources the Single Service Provider controls. In this environment a Single Service Provider(SSP) is any business that provides a telecommunication service, examples include Cloud providers, from IaaS to SaaS, Video Conference providers, to simple network connectivity providers such as the business side of telecommunication companies and ISPs.

The provider can entirely own their own resources or, lease resources from the Single Service Resource Provider(SSR). In the case of ISPs, for example the SSP would negotiate with the SSR for a certain percentage of bandwidth, where as for public Cloud providers it is generally the case that the provider has total access to the cloud environment.

The Single Service Provider maintains its own management system, from which it provides the BSA with monitoring / performance information on the contracted service. The nature of the Single Service Providers management system, with regards to centralised or distributed management, is not important to the scalability of the system requirements, since the coordination is done by independent agents. It is required that the providers' management system is capable of judging the performance needs of current and future contracted services, as well as be able to provide performance monitoring information.

3.5.1.4 Single Service Agent

The goal of the Single Service Agent in this environment is to maximise the utility of the service that it is providing, to that end the Single Service Agent (SSA) is responsible for negotiating the price of the service with the Bundled Service Agent and the Single Service Resources (as required). In the negotiation between the BSP and the SSP this includes providing quotations dependent on the services requested length. The SSA is also responsible for the scheduling of the contracted service.

Once the service is started the SSA interacts with the BSA via monitoring messages which are an amalgamation of the services internal monitoring messages, passed through the SSP own management system, that inform the BSA of the Single Service Resource status. The type of monitoring message that is sent is specified in the service contract. The information gathered for the status/monitoring messages is related to the current utilisation of the SSRs and the service performance.

3.5.1.5 Single Service Resource(s)

A Single Service Resource(s)(SSR) is any physical infrastructure service provider in the telecommunications environment. Examples include Telecommunication companies backbone network, Cloud Provider physical servers, and service provider content servers. In this environment the Single Service Resource Providers will be either wholly owned by the Single Service Provider, or leased on demand or over certain periods by the Single Service Providers. The Resource requirements include its own management system, which like the Single Service Providers management system should provide performance information to the Single Service Providers for scheduling and monitoring purposes.

3.5.1.6 Single Service Resource Agent

For service resources that are not owned by the Single Service Provider, the Single Service Resource Agent (SRA) is responsible for deciding on the resource's price, either on demand or over the SSP required time. It is also responsible for scheduling the contracted services in conjunction with the Single Service Agent. Finally, it exchanges performance information with the single SSP's management system through the SSA.

3.5.1.7 User Agent

The User Agent(UA) is responsible for negotiating with the Bundled Service Providers, in a request for quotation market. It is also responsible for any reconnection issues that affect the end user. In the case of negotiation the requirements of the user agent is to provide high level contract generation. For initial connection and connection issues, as this user agent will deal exclusively with its last mile provider, reconfiguration of the service during failure should rarely occur.

3.5.1.8 Service Information Management System (SIMS)

The goal of the Bundled Service Provider, and therefore of the Service Information Management System (SIMS), is to get the highest quality service for the user at the cheapest price which needs to be balanced against the Single Service Providers requirements for higher utilisation. Further to fulfil CARMA's goal of service responsibility, there needs to be an overall focus on risk management and mitigation for the service. For this to be possible the BSP requires a method of judging the risk of first, contracting with new providers, and second, what to do about the service under failure conditions.

3.5.2 New System Requirements for Risk Management

There are multiple factors that can influence the risk of contracting a bundled service. For example, if the user, situated in Australia, wishes to utilise a virtual machine in the east coast of America, in order to guarantee a quality service the bundled service provider is required to contract not only the virtual machine in the east coast American cloud environment but also all the network connectivity providers that exist between the user, in Australia, and the east coast American cloud environment. The risk then, of contracting this bundle is a combination of the risks of contracting with all the service providers in the bundle. Also affecting the risk decision is the quality requirements of the service, if the user has lower quality requirements for example, they are probably more accepting of a greater risk.

Taking the above example, if the user contracting the east coast cloud service, also wishes to utilise a video conferencing service in the west coast of America, then some of the connectivity providers between Australia and America could be shared. This sharing of resources for the two services in the bundle would lead to a decreased cost, but also an increased potential risk, as failure of the shared provider would result in the failure of two services.

The example discussed above shows that even a fairly simple case of single service provisioning introduces a number of variables that will have an effect on the resulting risk management decision. Effectively, the final risk assessment is dependent upon several components:

- Specifics of performance requirements requested by the customer
- Availability and characteristics of matching services
- Service-borne risks and costs of mitigation options
- Portfolio-borne risks and costs of mitigation options

The last point in this list refers to possible unforeseen interactions between actions that the bundled service agent takes to satisfy another customer contract, and decisions made to ensure availability, and levels of services required by the current contract. Such considerations are required in order to successfully manage a portfolio of services and customers.

3.5.3 Purpose of the SIMS

The Service Information Management System (SIMS) is responsible for judging the risk inherent in the management decisions made by the bundled service agent. At the highest level of abstraction, any risk management decision facing a bundled service agent (BSA), such as choosing the course of action in case of a noticeable drop in performance, or, providing a quote for a specific bundle request from the client, must take into account a multitude of variables drawn from a number of distinct information sources. Figure 3.3 provides a high level overview of the scope of information framework available to drive the Bundled Service Provider's decision making process.

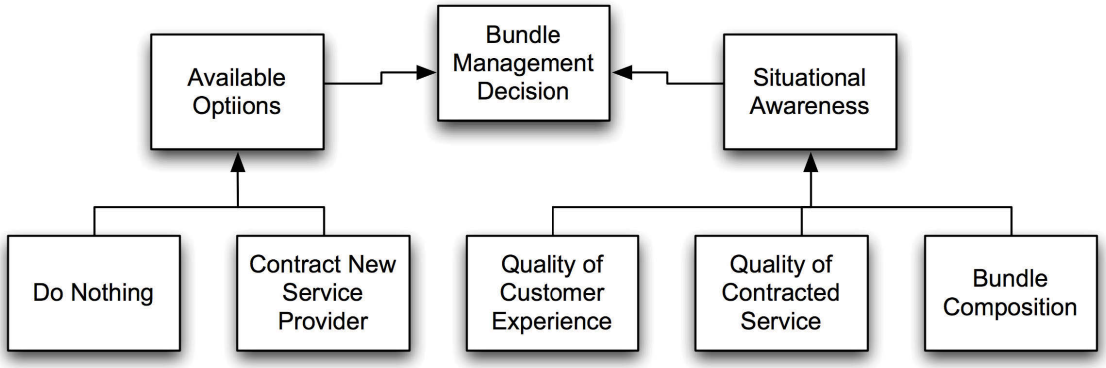


Figure 3.3: View of requirements and affecting factors of Risk Management.

As depicted, in broad terms, a decision is based on the assessment of what options are available and what their relative values are in a given situation with respect to the overall goal of management. Consequently, the mechanism of accessing information and determining the utility of options depends greatly on how one defines the BSA's management goal.

The above description represents the bundle management problem as one of portfolio management where the portfolio is of composed services with probabilistic estimation of benefits and costs. As such, the SIMS decision process can be defined as one aiming at risk assessment and minimisation, subject to cost constraints. As the Bundled Service Provider entity sits outside on top of the single service providers, and is a separate entity the choices of action are limited. Either the bundled service agent contracts with a new provider or it allows the service to continue. This leads to the following options:

- Do Nothing
- Instruct Single Service Providers to ease quality requirements for service continuance, or
- Contract a New Service Provider

Since the choice to change providers can be seen as a drastic action, it becomes important that the risk assessment that leads to this judgement is accurate. For accuracy in risk assessment to be maintained over time, it is important that the information it receives is continuously updated and evaluated.

Consequently the importance of the information on which the service information management system relies, is increased. It is envisioned by this system that as part of the contract with the single service providers, the single service providers would be required to send performance statistics throughout the lifetime of the service. The type of information that would be required by the Service Information Management System can be divided into three categories:

- Performance information from the providers
  - From which historical performance information can also be derived.
- User service quality requirements
- Service functional quality requirements.

However, should the service in question recover before the need to engage a fall-back materialises then the BSA can engage the fall-back service to fulfil a new contract and obtain unexpected benefits.

3.6 Assumptions and Risks of the CARMA System

The twin issues of security and trust are the primary assumptions and risks in the CARMA framework and are, for the specifics, beyond the scope of this work. In general however, with regards to trust, the author believes that the use of a market based system contains the necessary reward or punishment measures to enforce decent behaviour by the participants. For example, while the single service providers could choose to send false information regarding the state of the service running on their resources, if the service fails the Bundled Service Providers risk management would be less likely to use the service in the future. The converse of this is also true, with Single Service Providers with a history of reliability being more likely to be utilised in the future. As such, the author contends that specific rewards or punishments for Single Service Providers that negotiate in bad faith are not necessary.

With regards to the security of the services, this architecture and contracts between the Bundled Service Providers and Single Service Providers are compatible with current security technologies. This means that technologies, such as particular Virtual Private Networks (VPN) could be specified as part of the automated generated contracts.

Additionally, this work assumes that the SSPs have a management system which can provide both aggregated monitoring information and scheduling strategies to be able to provide real-time network performance information and predict future network use requirements. There is some concern that due to security concerns, the information that is required by the bundled service agent would be unlikely to be provided by individual single service providers. However, some wholesale divisions of telecommunication companies currently implement performance notifications for users of their carrier grade services through interfaces and sanitised performance information websites, Cloud environments providers also provide this information for their clients through such interfaces as Amazon's EC2 management console.

The primary method for resilience proposed by the architecture of the Bundled Service Agent in Section 3.5.1.2 is Renegotiation. This can be seen as a drastic step in service recovery. However, it is envisioned that this management of failure through negotiation would work in conjunction with the single service providers management systems. Modern cloud computing environments provide node replication services on failure and network management systems include the concept of redundancy in service provisioning. In this case re-provisioning then becomes the final step in managing failure.

3.7 Summary and Conclusions

In this chapter the requirements of the *Complete Autonomous Responsible Management Agent* framework were explored. The CARMA framework is designed to manage complex services across multiple domains and provide responsibility for the service. Responsibility is defined from a user perspective and is divided into two parts, that of resilience in the face of failure and risk management. This thesis presents CARMA as an augmentation of current network management and inter-cloud management systems. It is believed that this system, used in conjunction with current service management systems, would result in greater utilisation of current telecommunication networks and add responsibility, for the user, to any modern multi-domain service. To that end the requirements of the management system were defined and from those the initial elements of the system were identified. The bundled service concept was adopted as a representative complex service that utilises multiple providers in its fulfilment, and additional negotiation constraints were identified. The need for a market based system was proposed and market alternatives were discussed. The focus of these proposals was three exchange based sub-markets focused at differing layers of the system, the first being the primary market which is a request for quotation market, secondary, a clearing house for unused services and the wholesale market, which is focused on the interaction between the single service provider entities and the service resource entities.

The introduction of a market place is not a new concept for telecommunication services, however it has proved to be a difficult concept to implement. Until very recently business models in telecommunications have been primarily focused on the concept of static network services (Section 2.7). This simplifies their management as well as allowing business lead time for the provisioning of services, minimising the risk of implementation. Dynamic services however, with their minimal lead time, increases the complexity of the service and therefore the risk to the business. This risk has been a fundamental stumbling block to the implementation of a services market.

Despite this, the market place has continued to be proposed in research as well as implemented in certain sections of the telecommunications industry such as cloud computing (Sections 2.6 and 2.7). The author contends that while the reasons are varied, including greater utilisation of existing infrastructure, one large reason is the potential for mutual profit, on the part of the service providers as well as the potential savings for the users. This potential profit constitutes a very real business driver to accept the risks of complex dynamic services. However without a strategy for risk minimisation, the potential of a marketplace is yet unrealised.

Therefore the architecture of CARMA was developed and the roles and responsibilities of the entities was defined. The core entity, the Bundled Service Provider (BSP) is defined as a broker which negotiates for and maintains services in a bundle, providing *responsibility* for the service, through its management of the risk in contracting and re-provisioning and by providing resilience through monitoring, and re-provisioning of the service through its lifetime. The requirements for the Single Service Provider and the Single Resource Provider were discussed and defined in the context of telecommunication and cloud service providers and resources. Finally, a more in-depth definition of the responsibilities and knowledge requirements of the Service Information Management System (SIMS) was discussed.

The overall concept of *responsibility* and the architecture of the SIMS provides the strategy of risk management and minimisation that encourages telecommunication business to adopt a market based dynamic service environment. Additionally the implementation of resilience through renegotiation and contract specification provides a guarantee of service delivery which mitigates the responsibilities of the individual service providers and encourages their participation in the market.

The roles and responsibilities of the defined entities led to the system, which is presented in the next chapter. The next chapter covers the choices made in the design of the agents for the entities presented here, along with their intended interaction and functionality. The method of contract specification and transformation is defined, and the implementation of the SIMS is discussed.

Chapter 4

Model and Design

4.1 Overview

The previous chapter defined the entities of CARMA. This chapter will be an exploration of the issues, and design choices of the functionality of the CARMA entities and agents. The use of a multi-agent system allows current service providers to retain their current management systems. Initially the architecture of the CARMA is expanded to explain the functionality and responsibilities of each entity, with a discussion on the design choices of interaction between them for the purpose of ensuring responsibility through risk management and resilience. The design of the contracts and the translation between them is described, as well as the utilisation of a common information model and graph transformation. Finally an exploration of the utilisation of a Bayesian Belief Network is explored in the context of risk judgement as it applies to the performance of the Single Service Providers as seen from the viewpoint of the Bundled Service Providers and their agents.

4.2 Design and Functionality of the the architecture

4.2.1 Bundled Service Provider

From Section 3.5 the *Bundled Service Provider* is defined as a broker, that has the ability through its agents, to negotiate for contracts on the request for quotation market. The Bundled Service Provider also through its agents, manages the service throughout the service lifetime, managing the extra organisation constraints and dependencies that arise in the bundled service. Specifically, the Bundled Service Provider creates a Bundled Service Agent to manage the bundled service throughout its lifetime. For each agent, the Bundled Service Provider assigns a copy of the Service Information Management System that is responsible for the judgement of risk.

The Bundled Service Provider also maintains the database of Single Service Provider performance history that is utilised by the Service Information Management System. Furthermore the Bundled Service Provider performs the data mining tasks and any other analysis on the performance information for use by the SIMS. This analysis of the performance data is done in an asynchronous manner, and gives the Bundled Service Agent's (BSA) copy of the SIMS the information that is current at the time of agent creation. Figure 4.1 is a sequence diagram of the whole service cycle from the perspective of the BSP.

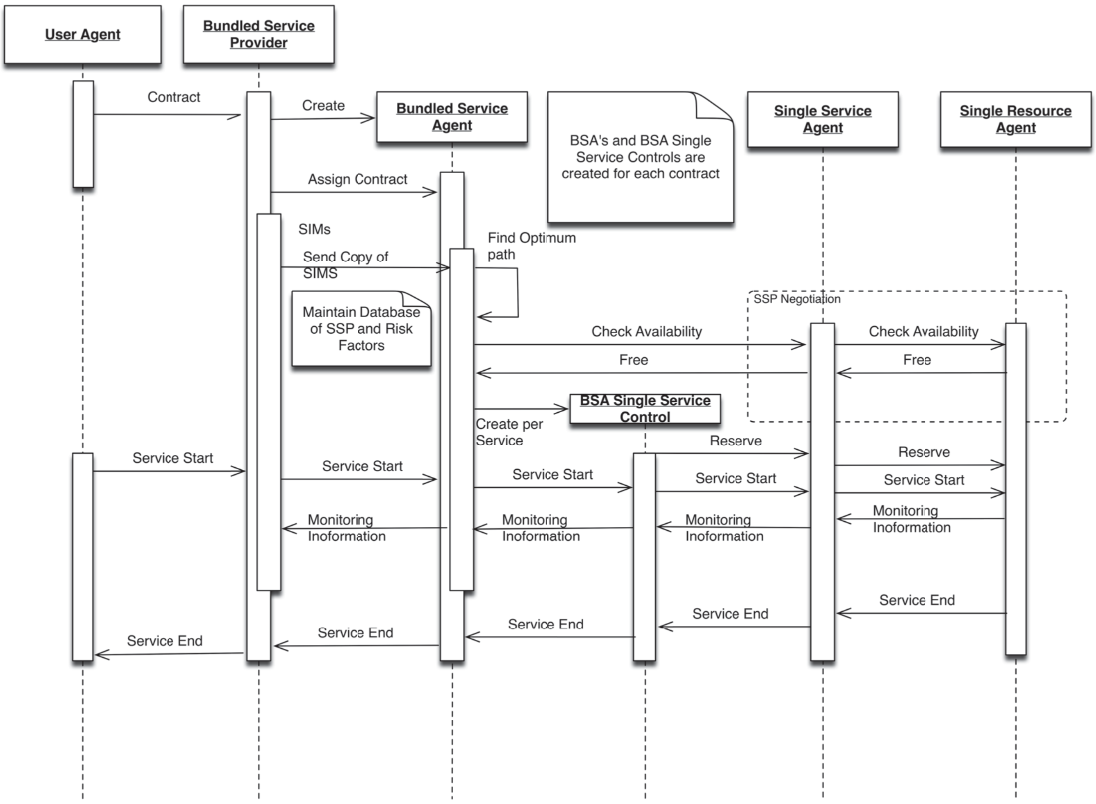


Figure 4.1: Whole Service Cycle from the BSP perspective.

A large part of the BSP's functionality is the management of the business aspects of the contract negotiation and transformation. Section 4.3.2 specifies the methodology of contract transformation from the high level user and BSP interaction to the lower level BSP to SSP interaction. However, the particulars of quantifying the requirements of specific services, such as the bandwidth required for a high quality Video Conference, is a business decision that is the responsibility of the individual BSP. The Bundled Service Provider is also responsible for the initial connection with the user agent, and for managing invoicing and payment.

4.2.2 Bundled Service Agent

The Bundled Service Agent (BSA) is the core of the CARMA system. As stated in Section 3.5.1.2 the role of the Bundled Service Agent is to manage the bundled service and provide responsibility for the bundle. Responsibility, as specified in Section 3.1 is defined through risk management and resilience. Specifically, the tasks of the Bundled Service Agent are: The BSA negotiates with the Single Service Provider's to obtain a quote for the service and manage the service throughout its lifetime. Figure 4.2 shows the state machine for the BSA. In this state machine, on receipt of a new contract the BSA moves to the Quote state, if the quote is accepted the BSA moves to the Manage Bundled Service state, if the service is not accepted the BSA moves to the No Quote state and finishes. The Manage Bundled Service state creates the individual single service modules for the BSA for the monitoring of the service throughout its lifetime. The states of the single service module are discussed in greater detail below. Once all of the individual services have finished, the BSA moves to the *succeeded* state, if one service in the bundle fails, and cant be recovered, the entire bundled service fails. Figure 4.3 shows the Bundled Service Provider sequence diagram for acceptance of a new contract. The Bundled Service Provider creates a Bundled Service Agent to manage the bundled service. The Bundled Service Agent then assigns single service modules to each service. These single service modules then negotiate on the market with the Single Service Providers to obtain a quote for the service.

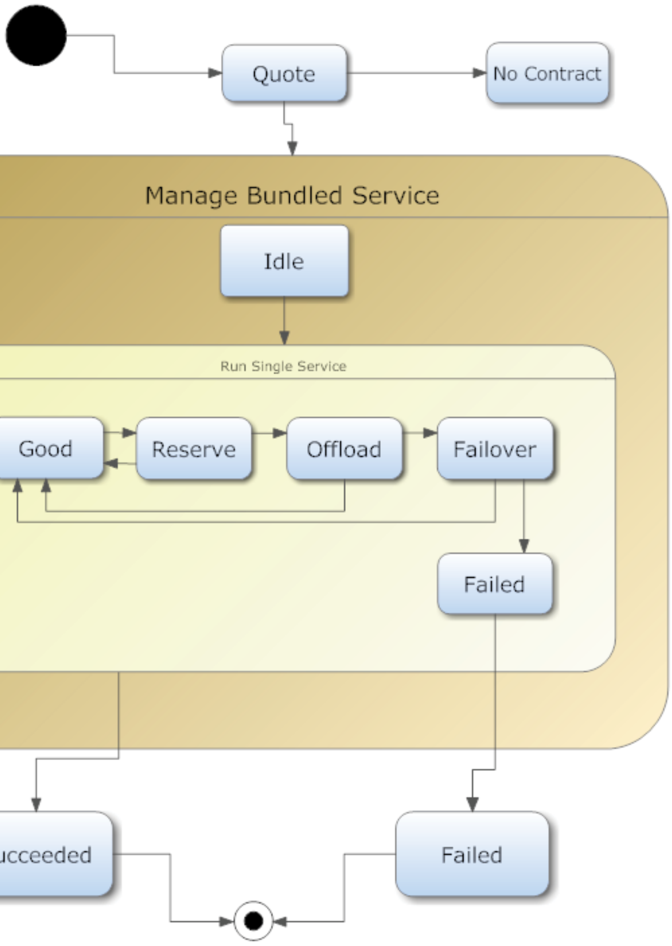


Figure 4.2: Statechart of the Bundled Service Agent with particular regards to the reprovisioning states.

The BSA's choice of provider is determined by its risk judgement of the providers as determined by the SIMS and the providers cost. The determination of provider is done through the utilisation of a Directed and Weighted graph. Graph Theory is utilised to determine the path of the bundled service. This is described in greater detail in Section 5.4.8. Once the contract has been negotiated and the services started the Bundled Service Agent creates single service modules and monitors the service through out its lifetime. The BSA judges the risk of failure and provides resilience through negotiation with new providers. In order to provide resilience from the BSA's point of view, it is necessary to receive monitoring information from the Single Service Agents with which it is contracted.

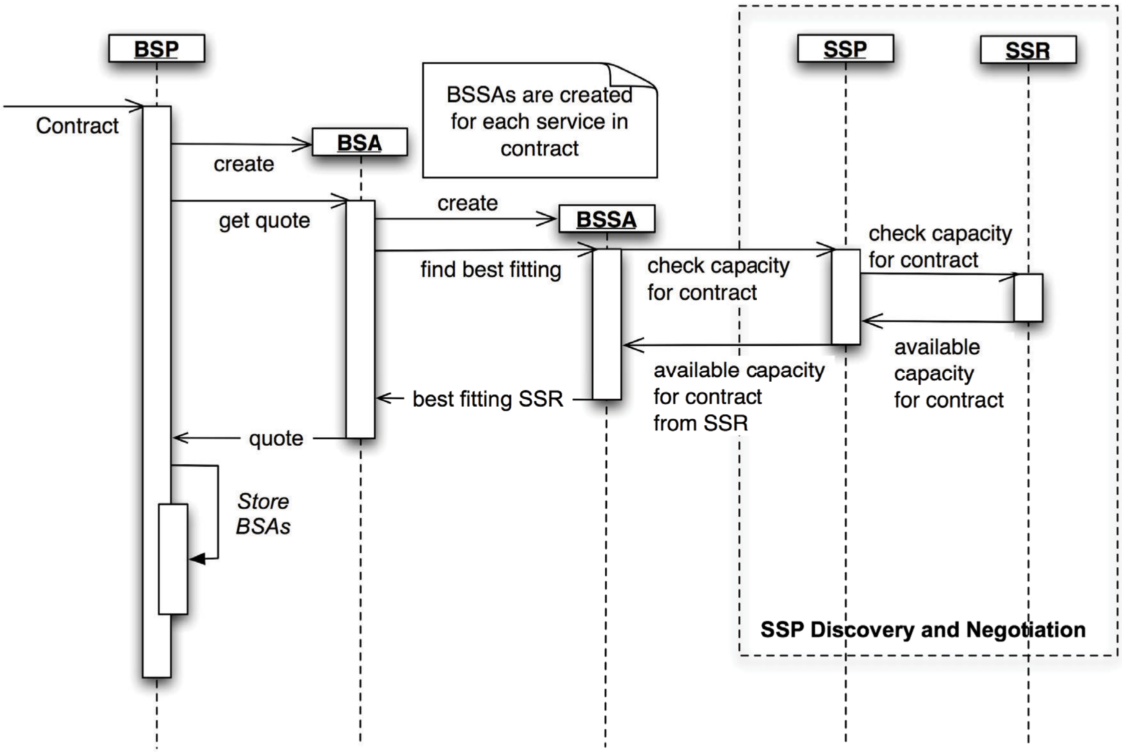


Figure 4.3: Sequence Diagram of contract creation and provisioning by the Bundled Service Agent.

For CARMA the BSP specifies in the contract that the BSA interacts with the Single Service Agents(SSA) through a series of service status messages, with these service status messages representing the monitoring information of the SSA on the service resources performance. The service status messages are gathered from the Single Service Resources raw data, typically SNMP, and aggregated by the SSA for transmission to the BSA's single service modules. In order for the SSA to classify the aggregated data the agents will utilise a table of thresholds, built from the SSP's management system. This translates such information as latency, for network connectivity services, or server response times in the case of virtual machines and applications, to overall performance data classified to four descriptions. These descriptions are green, yellow, orange and red, indicating whether the service is fine, degrading, in danger of failing, or failed. Current management systems for both network and cloud environments, such as Nagios, HP's Openview and Monsoon, as well as Amazon's EC2, which are discussed in greater detail in Section 2.7 utilise colour descriptions as a simple way to classify the status of the services that they are monitoring. While this colour classification is generally intended for quick human readable classification of faults, the principle of classification of service status is applicable to interactions across domains or service providers as it allows the depiction of a fault priority without revealing information that might be considered confidential by individual service providers.

The BSA processes this information through its copy of the SIMS to estimate the future service performance of the Single Service Provider. Based on the judgement of the SIMS the Bundled Service Provider moves through the single service modules control loop shown in Figure 4.2. The Control loop defines four actions states, that of *good*, *reserve*, *offload*, and *failover*. The action state *good*, defines the state when the service is performing at acceptable levels and no action is required. If the judgement of the services' performance level is too low, or that the information received from the SIMS indicates that the service has failed, then the bundled agent firsts instructs the SSA to drop the required level of service temporarily (to maintain some service functionality), renegotiates and *reserves* the service with another provider for the remainder of the services' time, starts *offloading* the service to the secondary provider and finally *fails* over to the secondary provider. For negotiation with the secondary providers the BSA again utilises the SIMS to determine the probable risk of engaging a particular Single Service Provider for service recovery.

Figure 4.4 is a sequence diagram of a failure scenario of Single Service Resource(SSR) and Single Service Agent(SSA) (a), with the renegotiation to a secondary provider and its resources (b). The SSR(a) sends raw monitoring data to the Single Service Agent, who processes the information, aggregates the data with the other resources that are controlled by the agent. The Single Service Agent in turn analyses the aggregated data and determines whether or not the aggregated data passes the predetermined threshold for a green, yellow, orange or red service status message. The SSA then sends the service status message to the Bundled Service Agents single service control module. The bundled services single service control module forwards this message to the SIMS which judges the risk of continuing the service with provider (a). If the SIMS determines that the service is degrading, then it sends a yellow decision message to the BSA's single service control module to negotiate and Reserve a replacement service with provider (b). If the SIMS determines that the service is in danger of failing then the BSA's single service control will send instructions to the new provider to start offloading traffic to the new provider. At this point the secondary provider is providing redundancy for provider (a). Based on the messages received from SSA (a) if the SIMS determines that the service is continuing to degrade, or has totally failed, then the BSA's single service control instructs SSA (a) to stop, and fails over completely to SSA (b).

4.2.3 Service Re-provisioning Composition

The simple sequence diagram Figure 4.4 describes the steps that the BSA would take in the event of failure. However, the choice of the provider for re-provisioning is also important

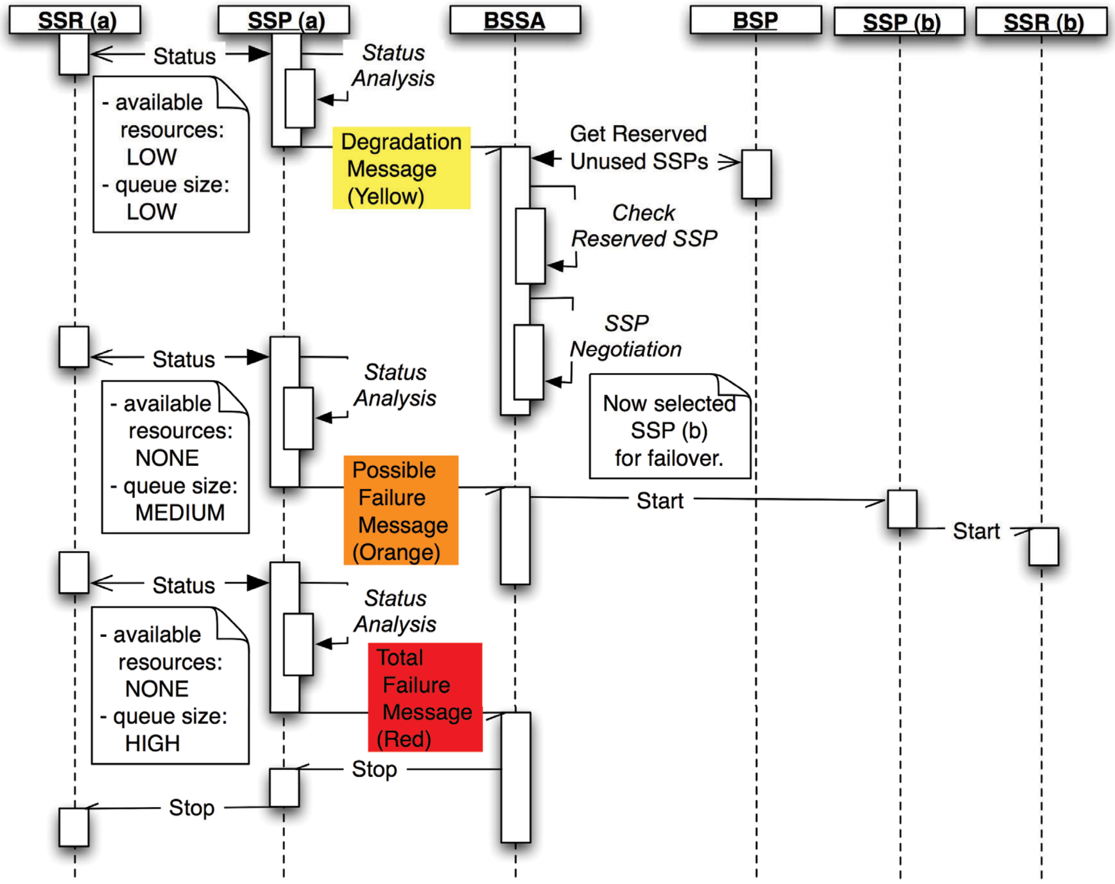


Figure 4.4: Reprovisioning Sequence Diagram.

and influenced by the number of providers that are needed in the re-provisioning path. Overall the BSA seeks to re-provision with the lowest number of providers to reduce both the cost and complexity of re-provisioning. In most cases that means re-provisioning with another provider in the same region. If that is not possible, the BSA would seek a path that minimises both risk and cost, to maintain the BSA's profit. An approach based on optimum paths is presented in the simulation in Section 5.4.8.

4.2.4 Single Service Provider

As stated in Section 3.5.1.3 a Single Service Provider(SSP) is any provider that exists in the telecommunication environment, from cloud Infrastructure as a Service (IaaS) and Software as a Service (SaaS) to Video Conference Providers and Telecommunication providers. In the market environment, they are defined as sellers of individual services that are required by the bundle.

It is assumed that the Single Service Provider implements some form of management system for its services, and that through this management system, and the business directives of the the provider, that the price of the service would be defined. For this system the SSP sets the price of access which is determined by the resource cost and the provider's total utilisation. The SSP agents implement a congestion pricing strategy which increases the cost of the service during periods of high resource utilisation. While this strategy of congestion pricing encourages BSA's to utilise other service providers, seems counter-intuitive, it has two main benefits. Firstly, during periods of high demand, the raised price barrier helps ensure that the risk to the performance of the already provisioned services is minimised. Secondly, higher demand returns greater profits.

For CARMA the Single Service Provider creates a Single Service Agent (SSA) that interacts with either the providers management system such as FOCALE or NetServ NAME which are autonomic or policy based management systems described in Section 2.4 or CLOUDBUS, SORMA, or Monsoon cloud management systems described in Section 2.5.1. The management systems would be used in the case of wholly owned resource systems. The alternative is interaction with the resource agent, in the case of leased resources. The SSA is also responsible for interaction with the BSA's single service module that is responsible for this service.

For the purpose of the CARMA system, there are three critical aspects of the providers management system, these are shown in Figure 4.5. The management system is required to be able to interact with the Single Service Agent to schedule accepted contracts, provision the accepted contracts and provide monitoring information on the service to the Single Service Agent in charge of the service.

4.2.5 Single Service Agent

The Single Service Agent (SSA) is the provider entity that is directly responsible for interacting with the BSA, the providers management system, and the remote resource agents. The main functionality of the SSA is to translate the agreed upon contract of the BSA to the providers management system, for implementation in the provider as well as translating the management systems monitoring information to send to the BSA. As such, while the agent is separate from the providers management system, the agent contains the ability to initiate a service, stop a service, and reduce the quality requirements of the service. As shown in Figure 4.5 The agent consists of three modules, for:

- Quotation
- Contract Transformation, and
- Monitoring system aggregation and classification

The quotation module works in concert with the contract transformation module and the management system's provisioning module to prepare a quote and contract for the service. The monitoring information aggregator receives performance information from the provider's management system and from the leased resource agent and sends that information to the BSA.

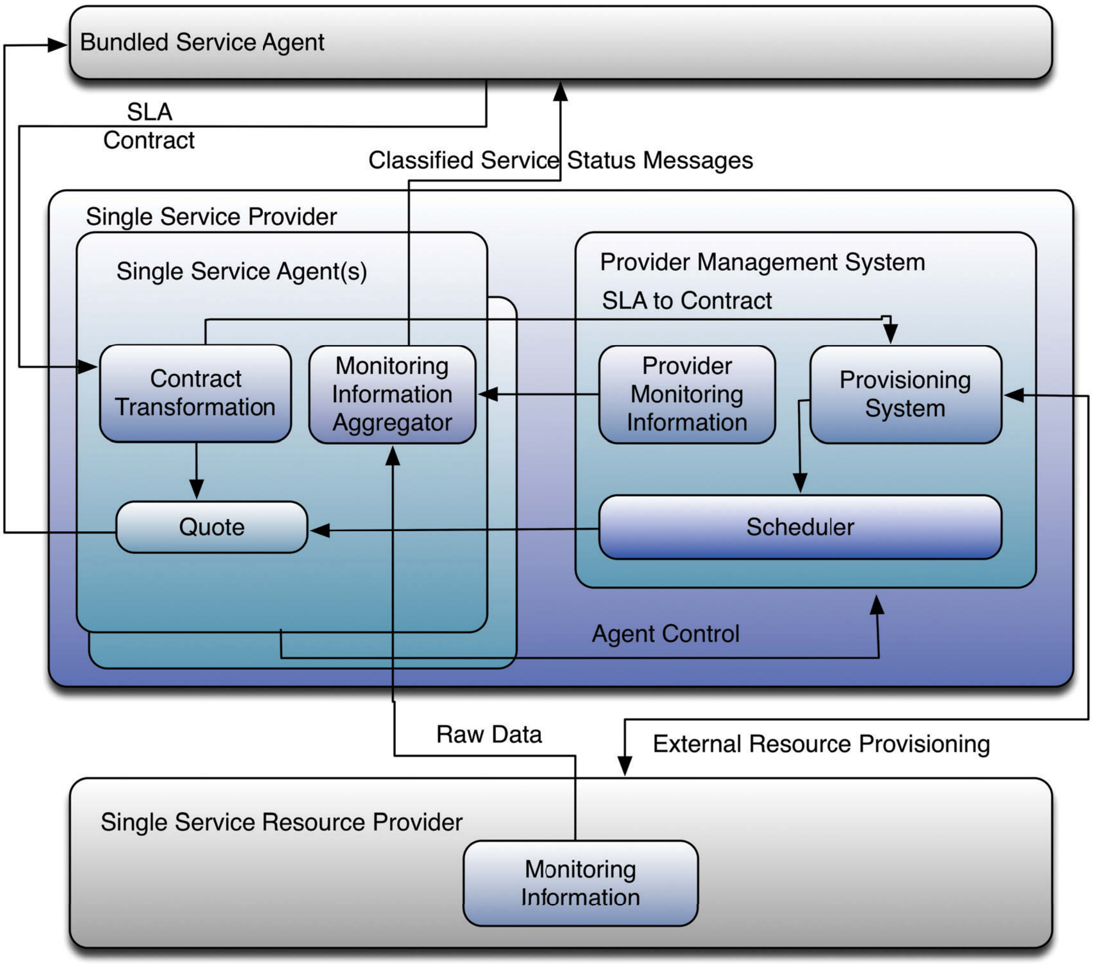


Figure 4.5: The architecture and functionality of the Single Service Provider and its agents.

4.2.6 Single Service Resource Provider

Taken from Section 3.5 the Single Service Resource Provider (SSR) is any physical collection of resources that a particular service is dependent on, for example, the servers for virtualisation in a cloud environment or the backbone network of a telecommunications provider. In CARMA the Resource Provider is responsible for setting the prices in the fixed price market, and for maintaining a management system that is either part of the Single Service Providers management network, or independent in the case of leased resources. Under lease conditions the SSP needs to communicate with the Service Resource provider through the use of the Single Service Resource Agent. The SSR's Management system communicates with the SSP through the Single Service Resource Agent.

As shown in Figure 4.6 the functionality of the SSR is similar to the functionality of the SSP, with the resource provider's management system providing scheduling, and provisioning of services that are leased dynamically, as well as providing the monitoring information through the Single Service Resource Agent.

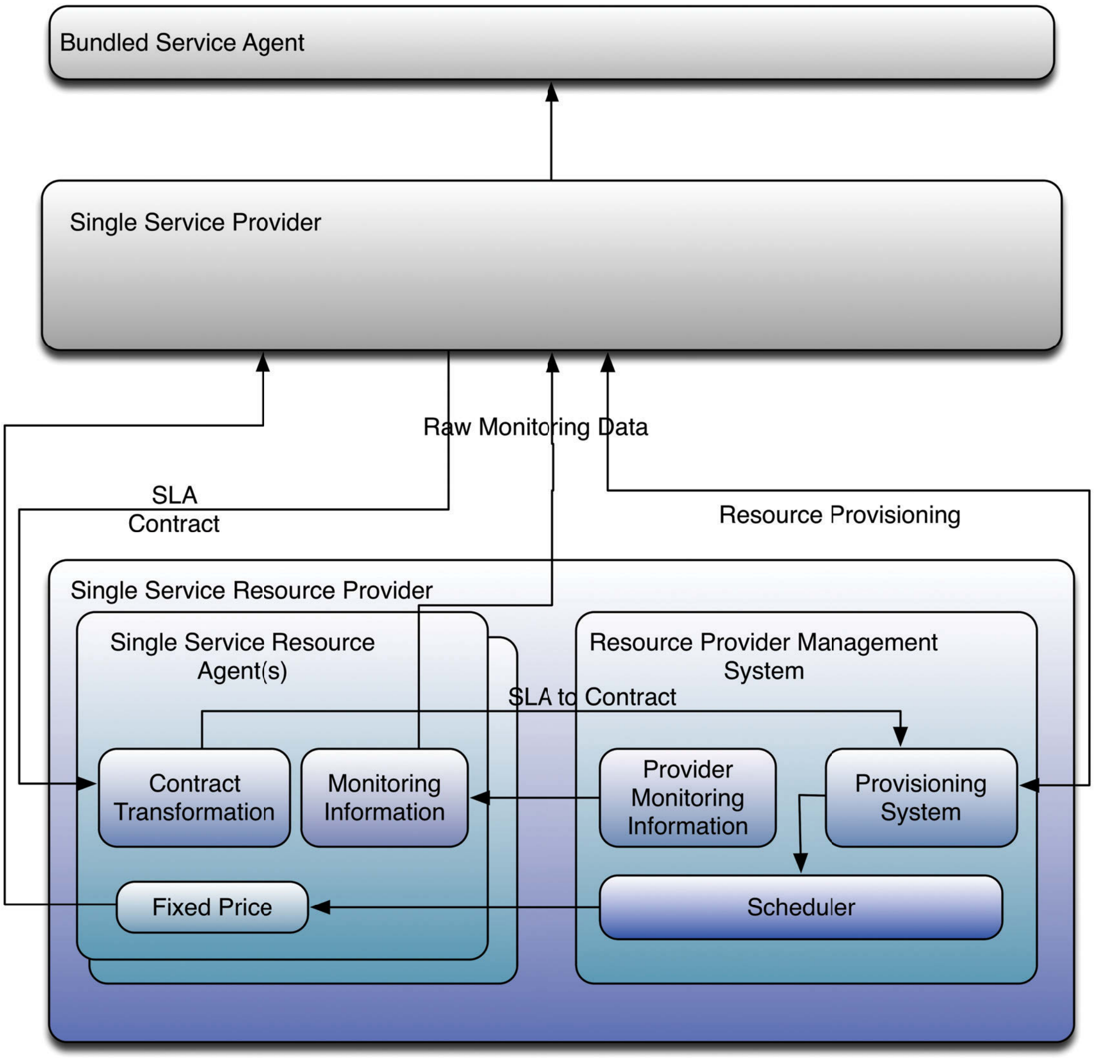


Figure 4.6: The architecture and functionality of The Single Service Resource Provider, its agents and its management system.

4.2.7 Single Service Resource Agent

The Single Service Resource Agent (SRA) is responsible for interacting with the Single Service Agent to accomplish the following tasks: provisioning the service on the SSR's resources, translating the monitoring information from the resources and changing the quality requirements of the service under failure conditions. Like the SSA, the SRA is used in conjunction with the SSR's management system and contains the specific information needed to translate the negotiated SLA contract information to specific resource control information. The interaction is shown in Figure 4.6.

4.2.8 Service Provider Composition

In this model, bundled and single services are composed of all intermediary resources providers that the service is routed through. For example a bundled service that connects a user to a virtual machine in a cloud environment involves contract that are generated for every telecommunication providers along the route, along with the end cloud computing environment. The basis for this idea of dynamic individual contracts with intermediary providers is based in the intelligent network concept proposed by TINA-C (Inoue et al., 1998), and uses the idea of a half-call model which was proposed as a alternative to the PTSN switch networks when IP and telephony were being integrated. In this version of the half-call model, the issue of global QoS is addressed by making the individual Single Service Providers responsible for the QoS in their own environment and the connections they control. Using individual and dynamic contracts allows failure to be traced more readily than the traditional static service subscription model.

The implementation of the contract's QoS remains the responsibility of the individual Single Service Providers and service resource providers. As the structure of the quality guarantee particulars of a High, Medium or Low quality connection are dependent on the type of service being provisioned. For example, for basic connectivity, quality can be ensured through technologies such as Multi-Protocol Label Switching (Rosen et al., 2001) and DiffServ (Blake et al., 1998). For the Cloud Computing environment quality is achieved through load balancing and extra virtual machine creation.

While the exact methodology of quality assurance changes for each service type. The particular quality requirements need to be specified in contracts that are verifiable across the multiple layers of the market place in the CARMA system.

4.3 Information Representation

For CARMA to be effective, the requirements of the contract specification shown in Section 3.3 described the need for specification that can cover the more abstract and simpler user specification and translate that information into contracts that contain a greater amount of specificity with regards to service requirements. At the same time representing the wide variety of knowledge and data that is required in contracting any service from multiple heterogeneous providers in a clearly understandable and interchangeable way, requires a common representation. A unifying language that can allow the different actors of the global market based management system to interact at three levels of specification, from abstract to concrete, from the user specification, the service specification and the resource specification. CARMA utilises both an already established information model and graph transformation to accomplish this goal.

4.3.1 Information Models and Ontologies

In the Contract Specification requirements Section 3.3 there is a requirement for semantic equivalency to be addressed when translating between different levels of abstraction and specification. One part of this equivalency is addressed through what Strassner calls 'mediation' (Strassner, 2003, p.112). Mediation is the use of a common information model to approach semantic equivalency in concepts.

Information models, which are commonly expressed in a modelling language such as UML, are based on the principles of object-orientation, and include such concepts as abstraction, and encapsulation which are of clear interest in specifications which move through different domains, such as are found in telecommunications and cloud computing environments. Similarly ontologies, a formal explicit definition of entities in a specific domain and their hierarchical relationships, domain semantics and entity properties, seems as well suited to the specification of contracts at the different marketplace levels.

However, with their principle hierarchical taxonomy that:

> a type S is a subtype of another type T if and only if all instances of S are also instances of T (Stenzhorn et al., 2008)

ontologies can also become inflexible in a global environment, due to the requirement that all properties of a particular subtype be met by all instances of that subtype. If a resource specification properties are extended for a particular resource, for example a network route, then for the network route type to be expressed unconditionally, all instances of the type must have the extended properties, which affects the total ontology.

Information models with their less rigorous definition allow for null value properties, which arguably is a more flexible way of dealing with a heterogeneous environment. However, this flexibility has drawbacks in that in any object in the information models domains, some properties are key. If they are not set, then generated contracts in the other domains could be invalid. CARMA uses graph transformation (see below) to provide the logical rigour to ensure integrity across the domains.

Information models have been created for a variety of industries. Two standard bodies currently offer information models for the telecommunication and cloud computing environments. The Tele-Management forums Shared Information / Data Model (SID in Frameworx) (Forum, 2012), and the Distributed Managements Task Force Common Information Model (CIM) (DMTF, 2011). Both are extensible frameworks, and cover resources, and services, to an detailed degree. The CARMA frame work has chosen to utilise SID because of its initial focus on telecommunication concepts for enterprise and service provider networks, as well as it's coverage of business and system viewpoints. TMForum's SID defines eight domains:

- Common Business entities: covering entities the span multiple domains
- Customer

- Enterprise
- Market and Sales
- Product
- Resource
- Service
- Supplier and Partner

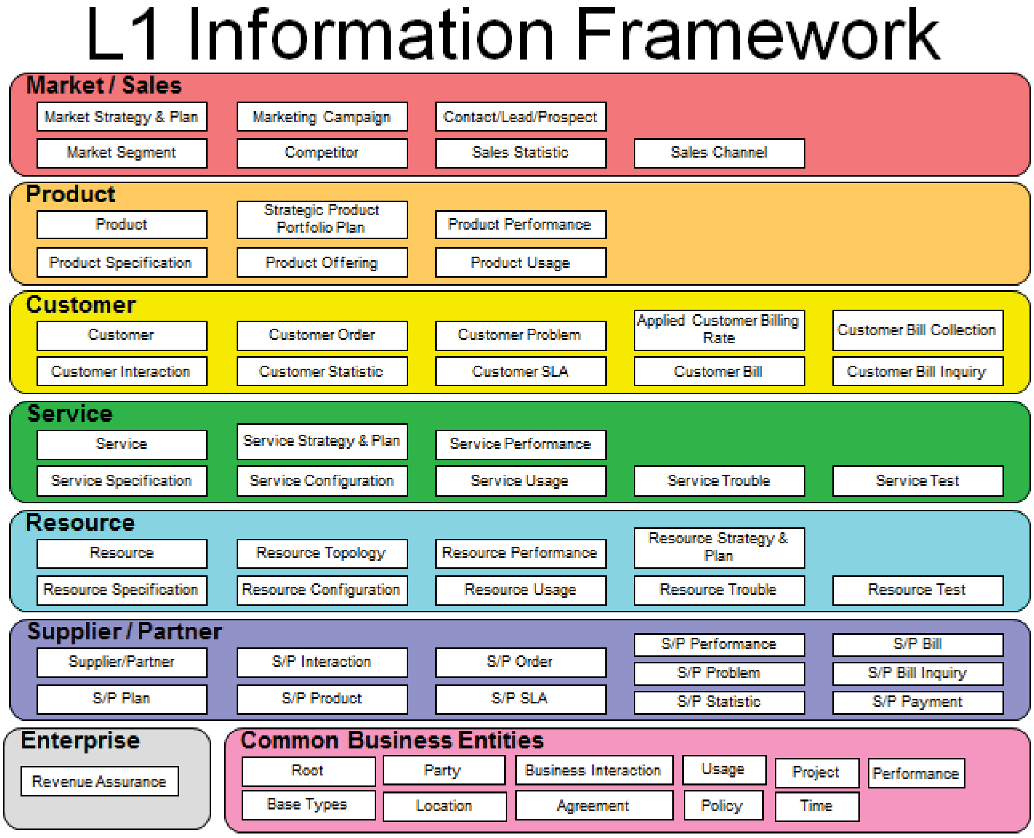


Figure 4.7: The TeleManagement Forum SID.

Figure 4.7 is a graphical representation of the SID domains. For contracting in CARMA over the three levels of specification, the framework makes most use of the customer, product, service and resource domains.

4.3.2 Contract Transformation through Graph Transformation

In terms of Service Level Agreements and Contract Negotiation, the various entities in this service model interact at different layers of abstraction. By this we mean that the information specified in the negotiations between the users and the Bundled Service Providers is more general than the information needed in the negotiation between the Bundled Service Provider and the Single Service Provider, or the Single Service Provider and their single resource provider.

For service providers such as cloud computing environments or telecommunication companies which wholly own their own resources such translation of details are accomplished by their management systems such as the cloud management systems or policy and autonomic systems discussed in Chapter 2. For the interaction at the user view or the provider view discussed in Section 3.3 ensuring validity happens through a process of adaption (Strassner, 2003, p.112) which provides semantic equivalency in contract syntax. There are multiple ways to ensure contract validity, from the semantics of ontologies to SLA templating. In the context of a system model that requires flexibility in an heterogeneous environment CARMA utilises a system of Graph Transformation.

4.3.2.1 Formal Definition

Graph transformation has been chosen as a means by which the varying layers of contract specifications are transformed and validated. What is required is maintaining equality in contracts that are specified in an abstract form that is suitable for a non technical user and a more specific form suitable for implementation by a service provider. In CARMA this was achieved by representing the existing contracts by graphs. Graph transformation can be accomplished in multiple ways, from the logical approach to the node replacement approach (Ehrig et al., 2006). For this contract specification the *algebraic approach* and the methods based on Category Theory called *pushouts* were chosen. Pushouts create sub-graphs based on the definition of graph transformation rules. These sub-graphs can then be identified in larger graphs and the rules applied, allowing the transformation to affect changes incrementally leaving the larger graph intact.

Graph transformation ensures consistency in the domain transformation, and allows structural analysis on the resulting graphs. Graph transformation has been particularly successful in the domain of architectural development (Denford et al., 2003), software engineering (Ehrig et al., 2006), and web service specification matching (Heckel et al., 2004).

4.3.2.2 Pushout-based graph transformations

There are two main types of pushout-based graph transformations in an algebraic transformation. They are the *single pushout* (SPO) (Ehrig et al., 2006, p. 14), and the *double pushout* (DPO). The significant primary difference between them is that fundamentally, a double pushout includes a *gluing condition* (application condition) (Ehrig et al., 2006, p. 11). It is this application condition that allows the transformation of contracts, to be unique if needed, by the various service providers.

The double pushout method uses the production rules $L \xleftarrow{l} K \xrightarrow{r} R$. In these production rules, L represents the pre-condition for the transformation, that is, what structure in the graph G must be present before the rule can be applied. The post-condition, R, represents how the sub-graph, represented by L, should be constructed after the transformation (Maxwell, 2007).

$$\begin{tikzcd}
L \arrow[r, "r"] \arrow[d, "m"'] & R \arrow[d, "m^*"] \\
G \arrow[r, "r^*"'] & H
\end{tikzcd}$$

(a) Single pushout

$$\begin{tikzcd}
L \arrow[d, "m"'] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "k"'] & R \arrow[d, "h"] \\
G & D \arrow[l, "\bar{l}"] \arrow[r, "\bar{r}"'] & H
\end{tikzcd}$$

(b) Double pushout

Figure 4.8: Pushout transformations in category theory.

In other words, for each application of the production rule, sometimes written as $r : L \rightarrow R$, the pre-condition graph L is sought in graph G. If it is found, then the mapping, m, defines the relationship between the nodes and edges in L, and their corresponding nodes and edges in G. The difference between the two graphs L and R is then applied to G, using the mapping m resulting in the output graph H. This transformation process is said to be commutative. That is, it conforms to (4.1).

$$L \xrightarrow{r} R \xrightarrow{m^*} H \equiv L \xrightarrow{m} G \xrightarrow{r^*} H \tag{4.1}$$

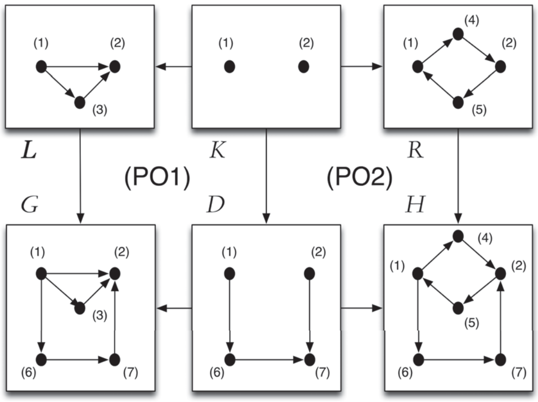


Figure 4.9: Graph Transformation Example. (Ehrig et al., 2006)

Figure 4.9 gives an example of a DPO graph transformation given a production $p = (L \xleftarrow{K} \xrightarrow{R})$ and a context graph D indicated by the general scheme in Figure 4.8. In the diagram pushout 1 (PO1), G is the gluing of the graphs L and D along K, with the graph morphisms being represented by the node numbers 1,2,3,6 and 7. Likewise in the diagram pushout 2 (PO2) shows the gluing of R and D along K resulting in H. This leads to the graph transformation $G \Rightarrow H$ via p.

When utilising pushout graph transformations, a prevalent issue that arises is what to do with the ‘dangling’ edge violations that can remain after a transformation. If either the source or target node of an edge is removed by a transformation and the edge and is not replaced by another node connection, that edge is described as ‘dangling’. There are two options for solving the violation of dangling edges. The more common is to invalidate any transformation that results in a dangling edge. The other option is to remove the dangling edge. This second option is less common as there is always the possibility that there is information contained in the edge that is lost when the edge is deleted (Maxwell, 2007). For graph transformation in the CARMA framework however, the information is contained in the nodes i.e. the service type and quality requirements of the service. It is therefore safe for contract transformations resulting in dangling edges to simply delete them.

One key point here is that graph transformation allows for the inclusion of properties that are not satisfied in all cases, meaning that the fulfilment of the contract by different service provider using varied but equivalent resources does not invalidate the higher abstract contracts, which is of particular interest in heterogeneous environments such as this hierarchical marketplace.

4.3.2.3 Usage of The Information Model and Graph Transformation

This section illustrates the usage of the information model and the graph transformation in the creation of contracts for bundled service negotiation. The required contract is defined by the user, using a template as blocks to build the required service through the use of an interface. This interface presents the requirements of the contract shown in Section 3.3 as generic choices for a bundled service. For clarity, the requirements are listed again as:

- Location
- Service Type
- Service Quality
- Start Time
- End Time

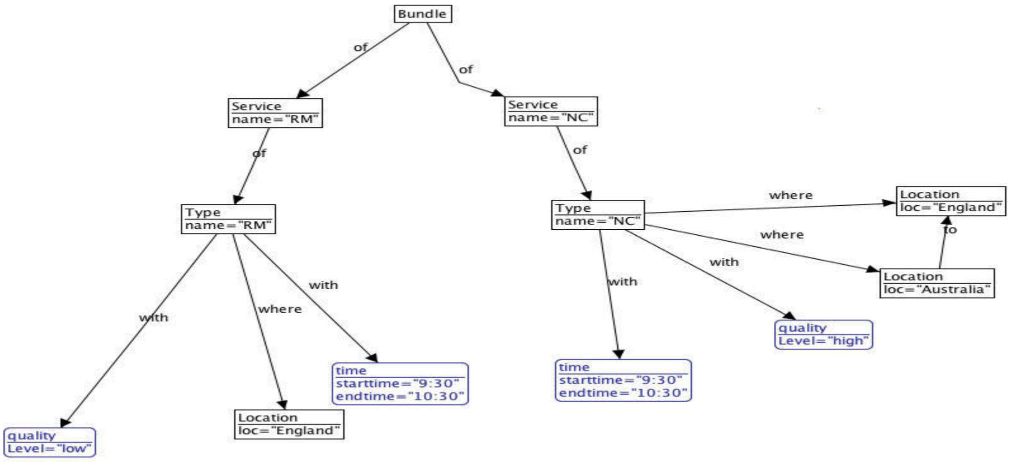
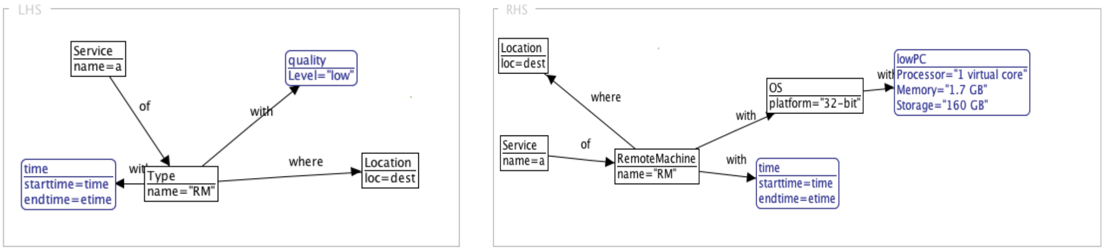


Figure 4.10: Untransformed Base Graph.

A representative example of a requested bundle is shown in the graph in Figure 4.10. The graph is displayed in the AGG graph transformation engine (Taentzer, 2004). The figure describes a generic bundle that is contracting a virtual machine in England and the connectivity between the user in Australia and the virtual machine in England for the set time of one hour taken from the type graph shown in Figure 4.17 (At the end of the Chapter). At the user level of specification the complexities of the connectivity and virtual machine are hidden, only requiring the user to specify the requirements in general terms. Figure 4.11 shows an example rule for the generic transformation from the user view to the service view, specifically for the Remote Machine or virtual machine service type. This example shows a suggestion of the choices the Bundled Service Provider would choose for low quality settings, (taken from Amazons EC2 cloud platform).



(a) Left hand side (b) Right hand side

Figure 4.11: Graph Transformation Rules.

Applying the rules leads to a transformed graph that specifies the services at a greater level of detail and is shown in Figure 4.12. The point of this transformation is that through a defined set of rules, set up by the Bundled Service Provider, can allow the provider to ensure that the abstract contracts can be represented in a level of detail that is suitable to negotiate with a Single Service Provider. Further this transformed graph can incorporate the specification of the TMForums SID to produce acceptable generic contract that can be implemented by heterogeneous providers.

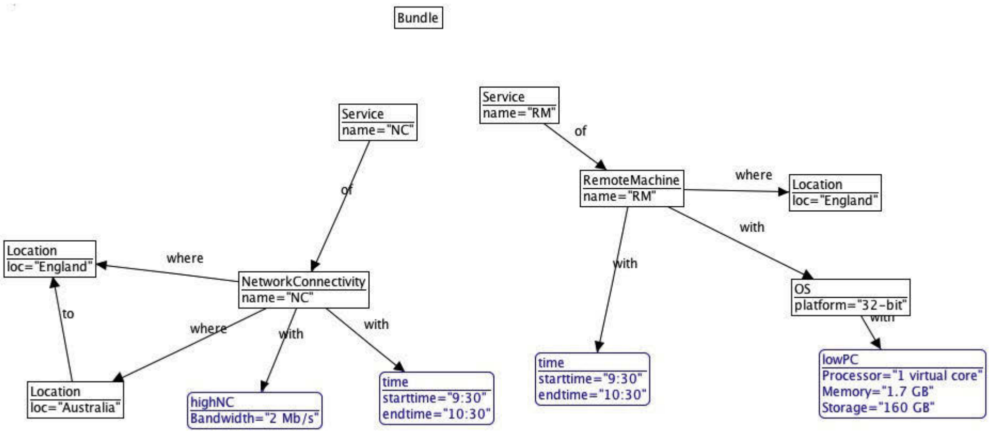


Figure 4.12: Transformed Graph into single services at the BSP - SSP layer.

To integrate the SID with the graph transformation requires identification of the Domains which the untransformed graphs and transformed graphs relate. Figure 4.18 (displayed at the end of the Chapter) is a UML (Rumbaugh et al., 2004) representation of a generic service specified across the Product, Service and Resource domains of the SID and outlines the classes necessary to specify the service across all three transformed graph layers.

Initially the User Agent would specify the requested bundled service as a Product Offering, which contains multiple products and their specifications. This Product and its specification shown in Figure 4.13, is only concerned with the Service from the users point of view only specifies the Service Type required, the Quality (specified as performance), Location and Time. These properties match the derived common properties of Section 3.3 (and re-listed above). The TMForum makes a useful distinction in the SID, that of a Customer facing service and a Resource facing service. Using the Customer facing entity for negotiation between the BSP's and the SSP's allows specification of services in a general and heterogeneous manner, without regard to the resources that comprise the service. Once the contract has been received and processed by the SSP, it is returned to the BSP with an appropriate price, and monitoring information attached.

For negotiation between the SSP's and their resources, if the resources are leased and not wholly owned by the SSP, the specification uses the Resource facing entity, which allows a more concrete specification of the resource requirements. Figure 4.14 is a very general UML diagram of the contract specification at this level.

The cross domain approach of the SID, coupled with the graph transformation done by both the BSP and SSP's (if required), together provide the rigour in translation between the contract specification levels. Allowing both adaptation of specified values and mediation between concepts that occur at the different levels of specification.

4.4 The Service Information Management System

The Service Information Management System is responsible for judging the risk of contracting a new bundled service, or maintaining the current service standard. To accomplish these goals it relies on information gathered by the Single Service Providers and its own information on those providers, as well as an understanding of the user quality requirements. The user quality requirements and basic performance information on the Single Service Providers, is information that can be agreed upon during the contract negotiation phase, while more in-depth knowledge of the Single Service Provider would be obtained by historically gathered and stored information about the Single Service Provider.

It is envisaged that the performance information gathered by the Bundled Service Provider would be in the form of traditional performance information alerts, such as SNMP, which has been aggregated by the Single Service Providers and is specific to the services non functional requirements. For example, the bandwidth of a connectivity service or processor

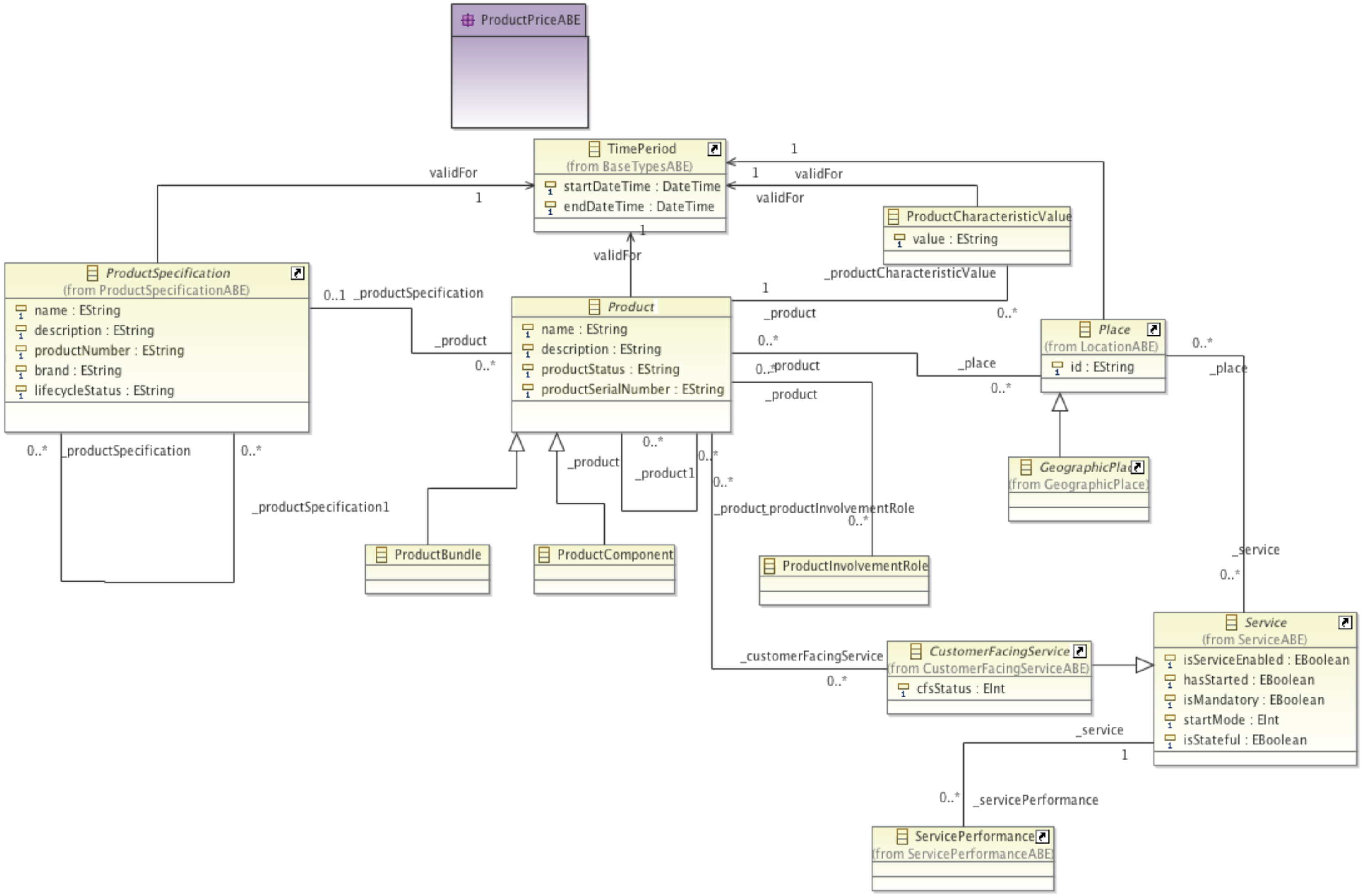


Figure 4.13: More detailed UML SID diagram for the contract at the BSP - SSP layer.

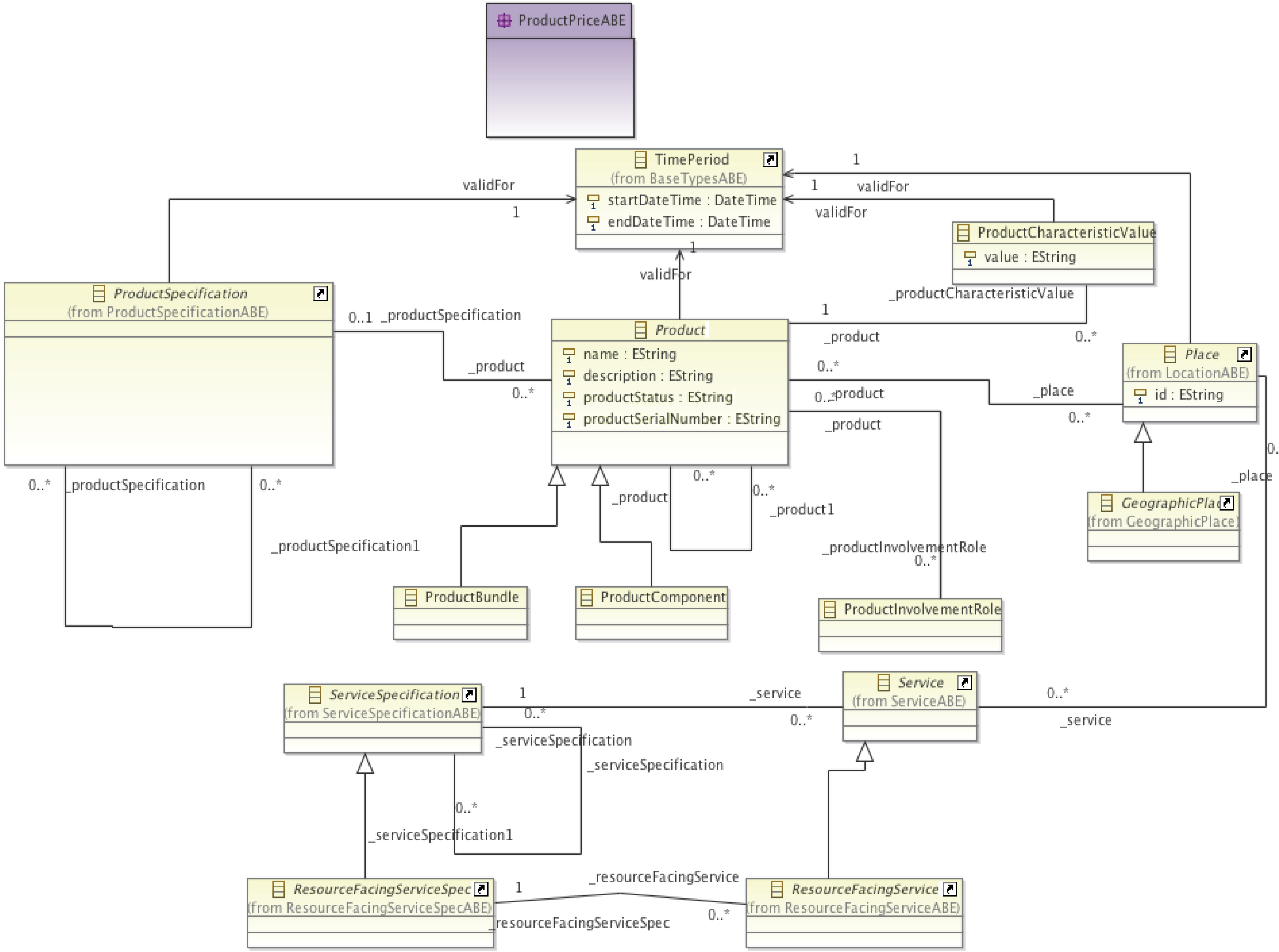


Figure 4.14: More detailed UML SID diagram for the contract at the SSP - SSR layer.

use in the case of a virtual machine. This information would be kept by the Bundled Service Providers that utilise the service and, over time, would utilise data mining techniques to interpret the historical information and provide a more accurate performance measure.

Utilising this information the Service Information Management System (SIMS) needs a mechanism for translating the vast variety of data into a recognisable risk assessment. For the first goal of determining the risk of contracting a new bundled service, the risk assessment can be seen to be strategic in nature, for while the calculation of risk includes historical information it is not reliant on a specific sequence of events. Rather the risk of providing the bundled service can be viewed as a combination of the various risks in contracting with all of the providers in this bundle. This calculation of the combination of risk variables lends itself naturally to the field of conditional probabilities. For this reason, the CARMA framework has chosen to use a dynamic Bayesian Belief Network (dBBN) for the risk assessment.

The second goal, being focused on calculating the risk of failure in one provider can be seen as a more tactical decision process as it is dependent on the sequence of status messages that is received by Bundled Service Agent. In this case using a dBBN is less effective as it becomes difficult to create conditional probabilities that takes into account the sequence of events. For this risk assessment goal, an algorithmic approach was devised.

4.4.1 The Dynamic Bayesian Belief Network in the SIMS

There have been various methodologies proposed for the assessment of risk in multiple fields. In practice most risk assessment is performed in a qualitative sense, utilising analysis of in place procedures and identification of risk factors. The difficulty lies in the identification of the needed weightings for the risk factors. Again in practice this is done manually by domain experts. As the system requires that the risk judgement is done automatically by the Bundled Service Agents, an automated approach is required. Two alternative approaches to Bayesian Belief Networks that have appeared in the literature uses neural networks or genetic algorithms (Song et al., 2006; Abdel-Aty-Zohdy et al., 2006). With neural networks however, problems arise from the uncertainty in the influence of the hidden layer of nodes. Additionally the time involved in the learning phase makes the solution impractical when dealing with risk assessment in an open marketplace that can change and introduce new services rapidly. The same problem applies to genetic algorithms, as it would have to be initialised and the selection phase run with every new change to the received performance information.

Existing research in the techniques of inference under uncertainty in telecommunication networks (Bashar et al., 2010) has shown that Bayesian Belief Networks (Na and Ping, 2010) can be used to draw qualitative estimations based on a large number of interacting variables. Furthermore, the BBN structure can be composed by combining known structural information with data mining techniques performed over historical data regarding service performance and reliability. As a result, this approach allows for a certain level of adaptability, which is desired given the potential for future changes in the nature of services and contracts that could be used to create the bundle managed by the autonomous agent.

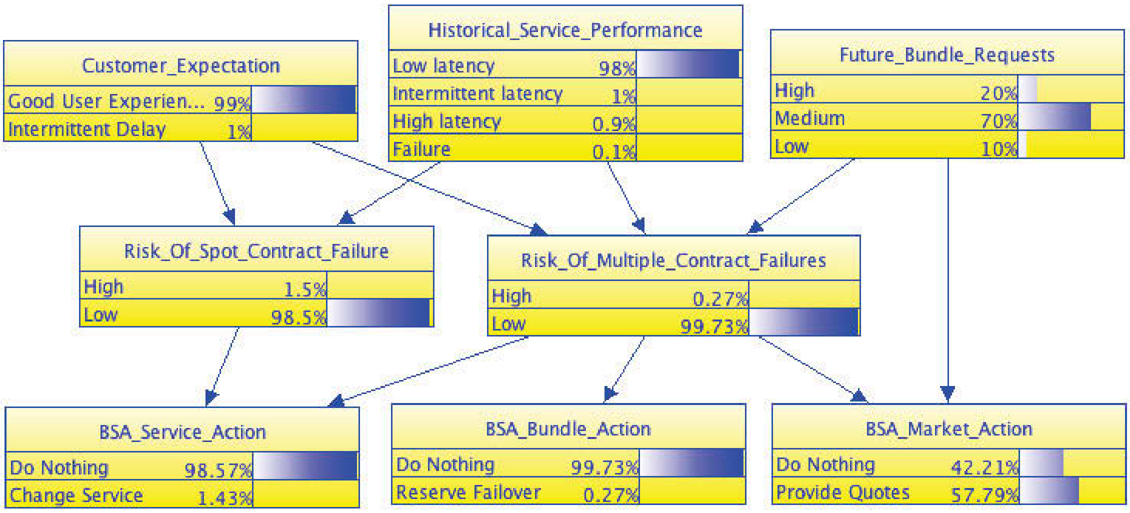


Figure 4.15: Bayesian Belief Network with no evidence for the purpose of Risk Judgement in Bundled Service Contracts.

Figure 4.15 depicts the initial attempt to construct a network which takes into account the BSA's position with respect to its current and future customers as well as its currently contracted services and attempts to convert a set of variables carried by this context into one of the three decisions: do nothing, contract a new service and obtain a new customer. Specifically, the nodes can be described as falling into one of three groups:

- Variable: these nodes are representative of variation in the BSA's operational environment.
  - Customer Expectation: estimates of performance based on customer information.
  - Historical Service Performance: aggregation of data from service's performance characteristics.
  - Future Bundle Requests: the SSPs existing commitments during the time frame of the proposed contract.
- Analytical: nodes of this type represent BSA's estimation of possible risk exposure.
  - Risk Of Spot Contract Failure: shows exposure to possible loss of income from a single contract which affects no other customers.
  - Risk Of Multiple Contract Failure: shows exposure to loss revenue from a suite of contracts which may potentially result in a significant loss.
- Actionable: provide the BSA with estimation of necessity for certain changes in the portfolio.

  - BSA Service Action: actions that affect one service within the bundle.
  - BSA Bundle Action: action that affects the composition of the bundle, in this case the BSA may decide to expand the composition of available resources.
  - BSA Market Action: represents BSA's behaviour in the market. At this level the BSA has only two choices, to provide a quote for the service or not.

Although simple in structure, this network shows the basic representation of the BSA's guidance dependent on ongoing assessment of current and future operational conditions. Naturally, with more historical data available one can discover more complex causality structures that would allow for more detailed representation of metrics, analytics, types and degrees of resulting action to be taken by the BSA. It is envisaged that going forward a variety of BBNs can be developed to typify BSAs targeting specific classes of bundles at different levels of risk. While specifically the historical service performance node calculates the risk inherent in contracting with particular SSP's agents, the remaining input nodes represent other influencing factors relating to the risk of accepting a newly proposed contract. The Customer Expectation node allowing some flexibility in provider choice risk if the expectation of service quality is varied. The Future Bundled Requests node expresses the increase or reduction of risk in contracting with specific providers based on the knowledge of the number of other contracts with that particular SSP. For example, the knowledge of a high number of existing contracts on the SSP relative to its capacity would increase the risk in contracting with that SSP.

As the project progressed through the action research cycles the composition of the dBBN and its use in the simulation changed. While this is explained in more detail in Section 6.4.2.1, in general the limited number of BSPs meant that for enough diversity in user choice in the marketplace the BSPs were required to quote on every contract. This meant that the output of the Market Action Node was unused in the simulation and that the fine tuning of the judgement of whether or not to provide a quote on every contract is left to future work. Additionally, as explained in the next section, the use of the Service Action node, the Bundled Action node and indeed the dBBN proved to be unworkable in the context of the service resilience judgement. These decisions are examples of how individual BSPs would alter and fine tune their own dBBNs in the context of providing the best optimisation for their users in a market environment, and the inclusion of the unused nodes is left to show the capture of the requirements for risk judgement described in Section 3.5.2. Despite these limitations the flexibility of Bayesian Belief Networks, with their ability to be dynamically altered to suit new circumstances, and be implemented without any lead or training times, like comparable neural networks or genetic algorithms, make their use eminently suitable to the prediction of risk in the dynamic telecommunication service environment.

4.4.2 Performance Judgement in the SIMS

The other requirement of risk management is the judgement of risk with regards to the probability of current service failure, described as the resilience judgement. For the resilience judgement, a BBN is not needed. While the BBN can determine risk based on the wide variety of causality structures, in this instance, the overall historical performance of SSP's will skew the resulting risk judgement of individual services during the service run. Failure of modern telecommunication and cloud computing services is generally measured in both the Mean Time Between Failures and the Mean Time To Repair, both of which are calculated as percentages generally ranging from 0.9990 to 0.9999 (the four 9's)(Arno et al., 2006), meaning that outages are registered in hours out of year, The reliability of the providers at all other times would falsely represent the potential for failure inherent in a service returning performance information messages classified as yellow or orange (as defined in Section 4.2). Therefore another, more immediate risk judgement of the services potential for failure is required.

```
for ( a sample of previous status messages ) {
  for ( each message of stated quality levels ) {
   if ( status message = green ) {
    reset problem predictor
   } else if ( status message = yellow ) {
    increase problem predictor by small percentage dependent on
 service type
  } else if ( status message =  orange ) {
   increase problem predictor by larger percentage dependent on
      service type
   if (problem predictor > failure threshold) {
  break
     }
 } else if (status message = red ) {
   problem predictor = failure threshold
   break;
  }
 }
}
 if (problem predictor >= failure threshold) {
   send failure / re-provision message
 } else if (problem predictor >= (service type dependent
high likely hood of failure)) {
      send immediate reserve / off load message
 } else if (problem predictor >= (service type dependent
     medium likely hood of failure)) {
    send reserve message
 } else {
    send ok message
 }
```

Figure 4.16: Performance Judgement Algorithm.

For the CARMA framework in the simulation this service failure potential is calculated algorithmically, dependent on the type of service being monitored (Service types are discussed in Section 3.3) and is shown in Figure 4.16.

The algorithm in Figure 4.16 describes an operator defined classification of risk weightings dependent on the type of status message sent by the SSP's and the service type. The failure, high and low problem thresholds, and problem predictor increases are estimated based on the authors previous experience with network based services and cloud services. For future work the author recommends investigation into support vector machines (Cauwenberghs and Poggio, 2001).

4.5 Summary

In this chapter the design of CARMA has been expressed, with the functionality of the entities in the system being initially described in Section 3.5. The design decision made in the context of interaction between the Bundled Service Providers, Single Service Providers, Single Service Resource Providers, and all their agents was explored, focusing on the structure that is designed to provide the top level responsibility for the service. To this end the chapter describes the sequence for negotiation between BSA's and SSA's as well as the steps required for renegotiation, specifically the four states of Good, Reserve, Offload and Failure, which represent the four stages of renegotiation. Also specified were the four classifications of performance information, being green, yellow, orange and red, which indicate whether the service is ok, suffering problems, in danger of failing and failed respectively.

The design of the CARMA entities and their interaction definitions have been, in some respects, kept deliberately abstract with the use of four catch-all categories of failure and only two stages of service failure resolution. At the level at which CARMA operates, the individual telecommunication service providers require a degree of autonomy and heterogeneity in provisioning and reporting of service status and performance. It is expected that in certain respects the interactions would be individual to the service provider and specified in the provisions of the contract.

Therefore for completeness, in the CARMA design this chapter then covers the mechanics of contract specification and translation across the levels of interaction in the marketplace, utilising the information model devised by the Tele-Management Forum and graph transformation. The use of the Tele-Management Forum's information model gives the contract negotiation in CARMA a solid foundation in industry based service specification, while the use of graph transformation for the automatic translation and specification of abstract service contracts provides a level of flexibility that is lacking from the static template alternatives.

Finally the design of the Service Information Management System of the BSP and agents is defined, utilising a Bayesian Belief Network for contract risk calculation and an algorithmic approach to performance degradation risk.

All of the elements of CARMA that were outlined in this chapter were designed towards scalability. The CARMA model, as outlined in this chapter contains no centralised components or knowledge plane, with each entity in the design being distributed and autonomous, for the length of the service in the case of agents, or for the length of the simulation in the case of providers. All cooperation was agreed through contracts and the marketplace, with no one entity being dependent on another, even in the case of multiple agents on the same resource, limiting the bottlenecks that can occur even in distributed designs. The knowledge plane represented by the SIMS was designed to be replicated across the Bundled Service Agents and with its focus entirely on risk management, realistically reduces the knowledge needed to make decisions and the space and processing power needed to implement and search this knowledge plane.

The next chapter covers the design implementation in the simulation, specifying the method used to implement the design discussed in this chapter, including the action methodology phases as applied to the simulation. Also introduced is the case study upon which the simulation was designed.

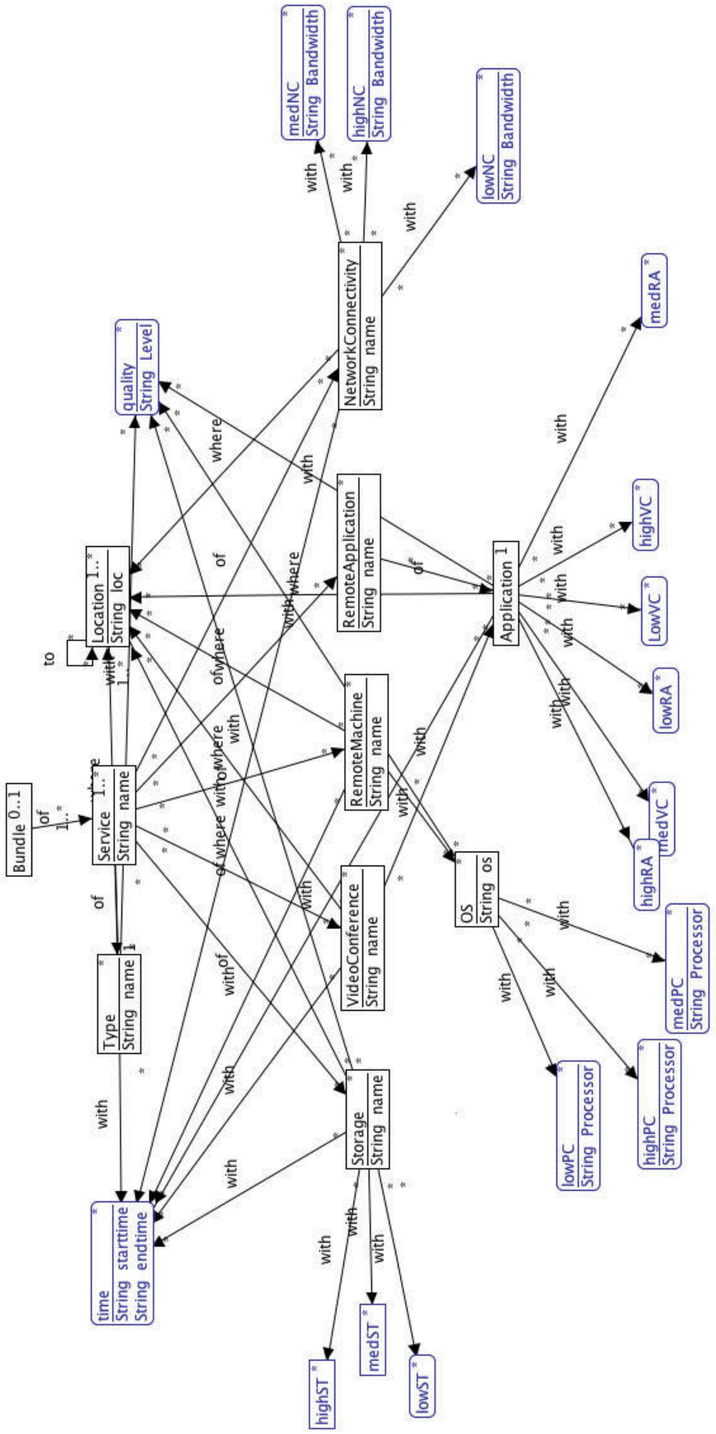


Figure 4.17: E-Type base graph of Transformations.

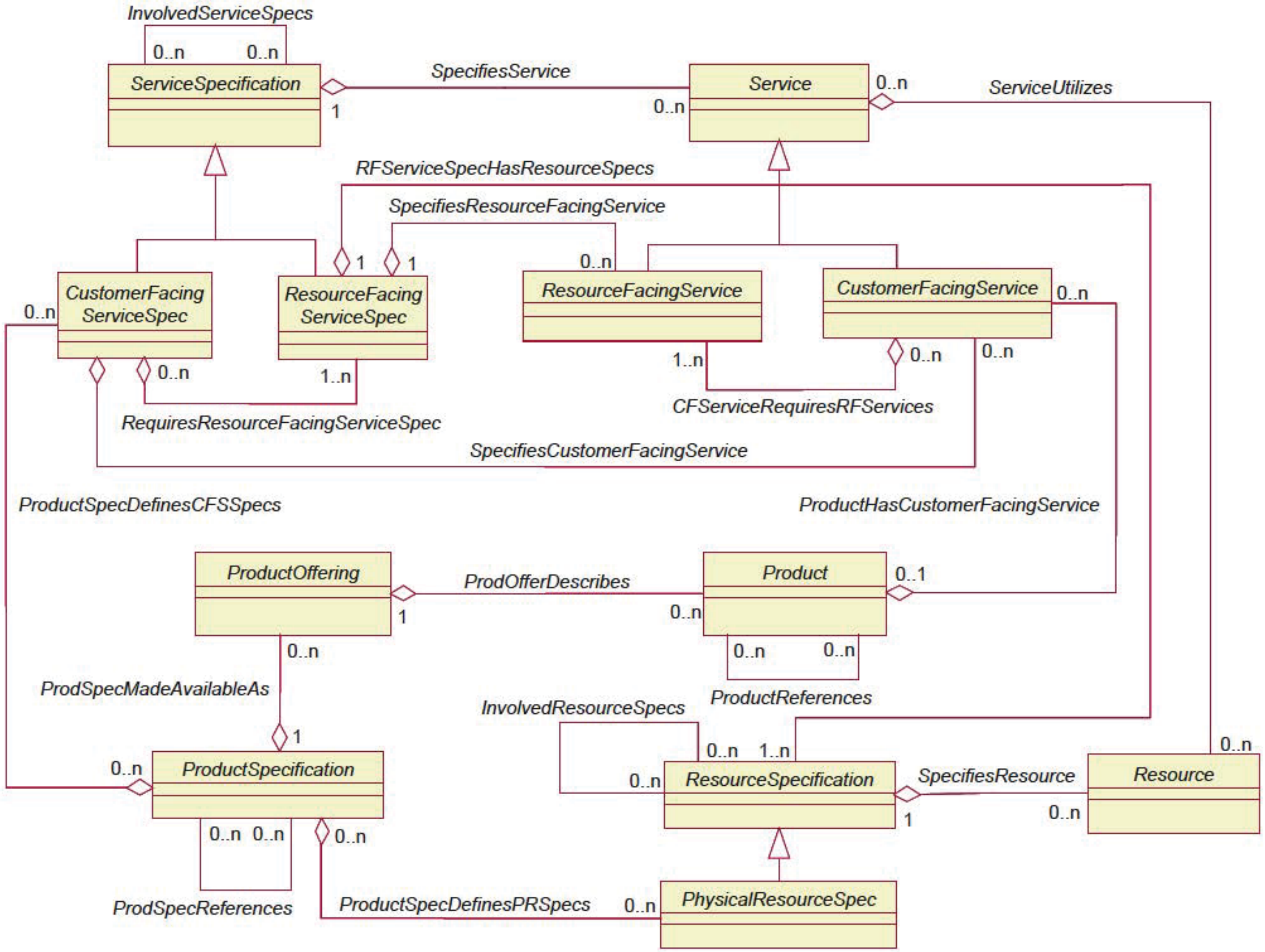


Figure 4.18: General UML diagram of the SID for a service across all layers.

Chapter 5

Simulation

The focus of this chapter is on the decisions, assumptions and constraints of the experimental simulation that was created to explore CARMA's system and design as it is described in Chapters 3 and 4. The simulation was created in concert with the design, following the Action Research methodology outlined in Section 1.3, evolving through the cycles as the architecture and design did. The structure of this chapter follows the outcomes of these cycles and simulation design changes, presenting first, the initial simulation design decisions with regards to the type of simulation model as well as the focus of the simulation and the level of detail required. Secondly this chapter presents the initial simulation implementation and exploration, including the choice of agent modelling language, and justifications for its use and suitability and ultimately the reasons for the rejection of this simulation choice. Thirdly, the alternative approach is presented with an discussion on equivalence, leading to the presentation and exploration of the final implementation of the CARMA entities and agents in the simulation. Lastly this chapter presents a case study which formed a basis validating of the simulation design and the nature of the contracts and traffic that would be simulated and evaluated.

5.1 Traditional Network Simulation

One approach to understanding and managing networks is simulation. Traditionally this modelling and simulation is done at a low level focusing on the devices that comprise networks and cloud environments. For example OPNET (Chang, 1999) is a simulation tool that can define in detail the various devices, protocols and technologies that make up a network. Containing various simulation tools OPNET provides a system administrator with the ability to predict end-to-end QoS with a high level of fidelity on a range of alternate designs using the networks technology. However, the focus on modelling devices and protocols naturally limits the viewpoint of the designer. One can see with remarkable accuracy the theoretical throughput of a particular router but one can not view how this bandwidth limit can affect the performance of a distributed service which is utilising the router. However, for the purpose of end-to-end services in a global market place, there is a requirement to move towards a more holistic representation of the network and its services for management purposes. One which encompasses not just the system administration viewpoint but also the various abstracted viewpoints from business to users.

5.2 CARMA Simulation Requirements

For this approach to management modelling and simulation to be effective the CARMA simulation is required to model: the processes of quoting, risk evaluation at both the initial contract negotiation and against failure, single service negotiation (including re-negotiation on failure), and service completion and failure.

Individually, the modelling requirements are:

- Process modelling,
- Constraints on state transition, and
- Goal based modelling, with explicit failure and abort states.

5.3 Agent Based Simulation

The system as designed in Chapter 4 is agent based, with the Bundled Service Provider utilising agents to negotiate and manage the contracts from users across multiple network and cloud service providers. Since the basis of CARMA is agents, it is reasonable to use Agent based simulation to simulate the system as opposed to equation based modelling. The simulation was built using the Java implemented agent based AnyLogic toolkit (XJTechnologies, 2010). Agent based modelling provides a *natural* view of the system, allowing the modeller to simulate the behaviour of the individual entities and their relationships in the system. It is particularly effective when the behaviour of individual entities is complex. In the case of CARMA, there is a high level of complexity in the interactions between the Bundled Service Agents, the Single Service Agents, the User Agents and the network/cloud Single Service Resource Agents. This complexity defies easy attempts to aggregate CARMA's entity behaviours, and using equation based modelling would increase the number of equations required to simulate the system exponentially.

Through the individual agents complex relationships *emergent* behaviour can occur and be captured by an Agent based simulation. Emergent behaviour is the concept that intelligent behaviour:

> emerges from the interaction of simpler behaviours' (Hall, 2001, p.85)

Emergent behaviour can not be predicted through the analysis of the individual elements of a system, rather can only be judged through analysis of the behaviour of the system as a whole. (Li et al., 2006)

Agent based simulation focuses on the entities activities, rather than business defined processes that are used to fulfil the system goals. Finally agent based simulation is flexible and allows the simulator to explore permutations much faster and easier that equation based simulation.

5.4 Design of Simulation Agents

5.4.1 Petri-nets

Initially coloured Petri-nets were used to design the control structure of the agents. Coloured Petri-nets, which are an activity-based modelling language (van der Aalst, 1998), are used for high level agent modelling. In activity-based modelling, the activities all have a goal and the termination of the activity always achieves the goal. Modelling intelligent agents requires modelling the goals that the agents work towards.

For the purpose agent based modelling, the goal defined in a Petri-net may only be a sub-goal of the agent. The multiple sub-goals will be managed by the layered Petri-nets used in the simulation. In our case, Petri-nets, combined with the programmable simulation environment (AnyLogic) initially provided flexibility in agent modelling.

A non-hierarchical Coloured Petri-Net definition, (taken from Jensen (2009)) is of a nine-tuple $CPN = (P, T, A, \sum, V, C, G, E, I)$, where:

- P is a finite set of places.
- T is a finite set of transitions T such that $P \cap T = \varnothing$
- $A \subseteq P \times T \cup T \times P$ is a set of directed arcs.
- $\sum$ is a finite set of non-empty colour sets.
- V is a finite set of typed variables such that $Type[v] \in \sum$ for all variables $v \in V$.
- $C : P \to \sum$ is a colour set function that assigns a colour set to each place.
- $G : T \to EXPR_V$ is a guard function that assigns a guard to each transition t such that $Type[G(t)] = Bool$.
- $E : A \to EXPR_V$ is an arc expression function that assigns an arc expression to each arc a such that $Type[E(a)] = C(p)_{MS}$, where p is the place connected to the arc a.
- $I : P \to EXPR_\varnothing$ is an initialisation function that assigns an initialisation expression p such that $Type[I(p)] = C(p)_{MS}$

5.4.2 Initial Bundled Service Agent Design

The Petri-net modules for the single services model the behaviour of the single service from the view of the Bundled Service Agent. The initial design utilised a broad token value set, with tokens representing service providers as well as service status, service usage, and service termination. Further, the Petri-nets constraints represented the constraint operators for each particular single service, such as time, cost and minimal allowed quality and whether the service is active.

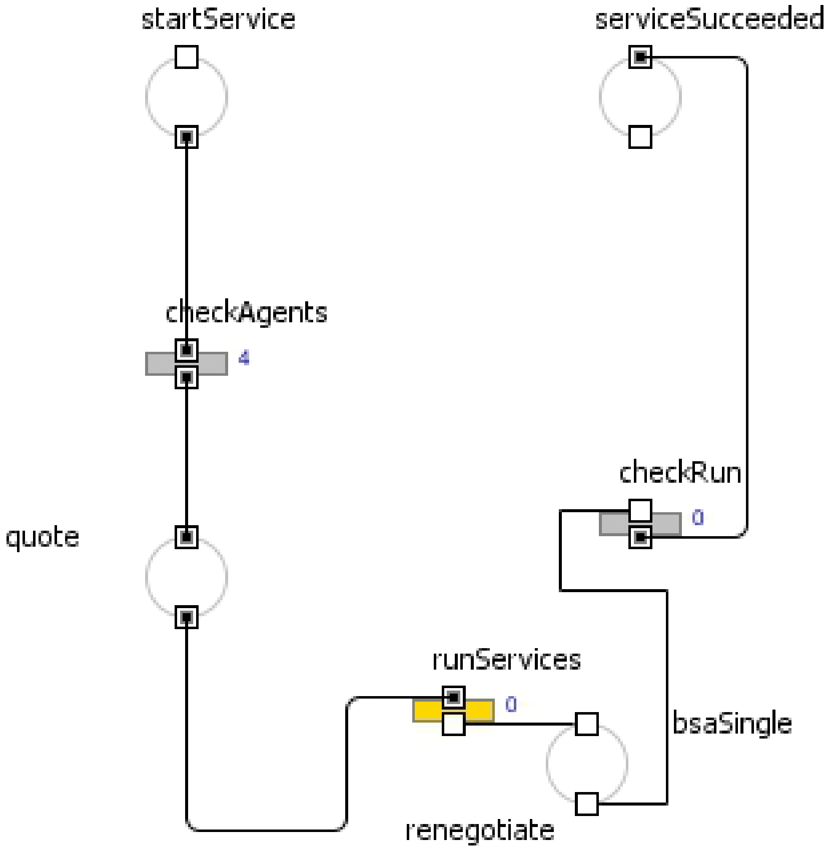


Figure 5.1: Initial agent control devised using petri-nets, top layer.

Due to the nature of modular, or hierarchical, Petri-nets, and the presentation limitations of the Anylogic modelling tool, the full structure of the Petri-net is not visible. Therefore Figure 5.1 shows the main control Petri-net of the Bundled Service Agent for negotiation (quotation) and execution of bundled services. The initial marking of the Petri-net has tokens representing the individual services in the startService place. The Petri-net seen here is added to depending on the number of services in the bundle, with the runServices transition and bsaSingle place representing the Petri-net shown in Figure 5.2.

Figure 5.2 represents the control flow of the Bundled Service Agent for interaction between the users and the Single Service Providers. With an initial marking of provider tokens in the createServiceProviders place, the start transition startNextProvider is fired when a token is placed in the renegotiate place from the runServices transition in Figure 5.1.

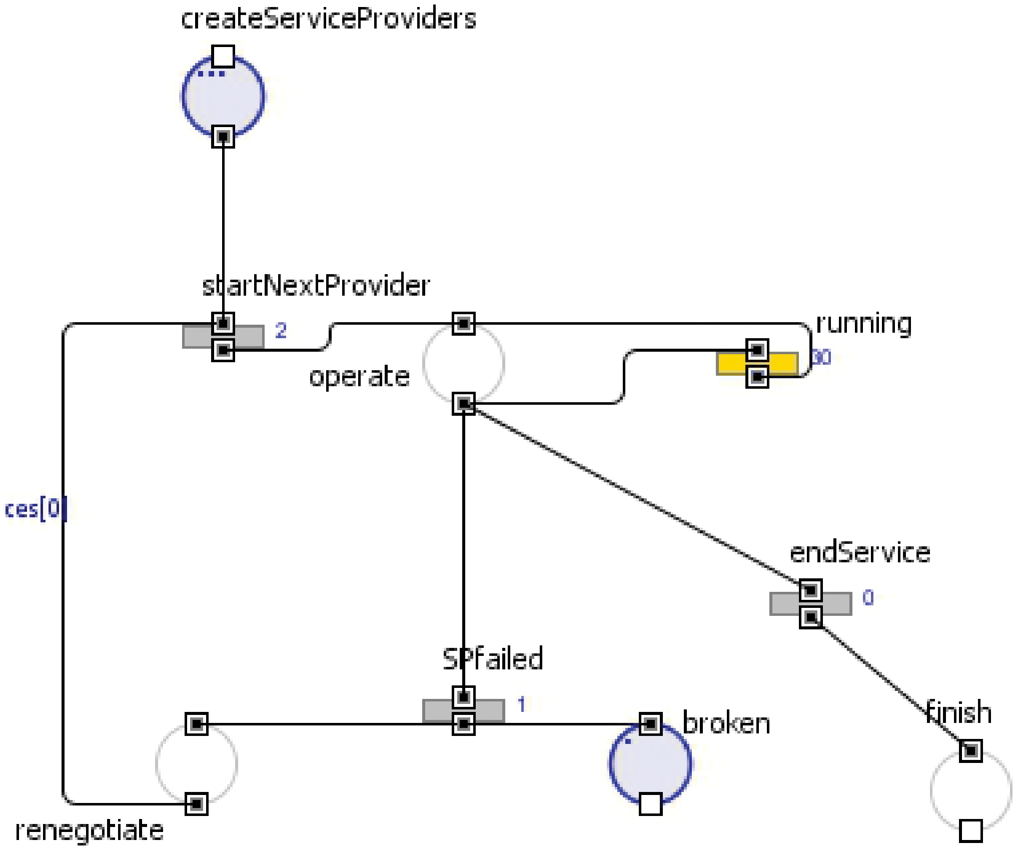


Figure 5.2: Initial agent control devised using petri-nets, bottom layer.

While Petri-nets were very useful in initially simulating the control behaviour of the agents, as the complexity of the simulation grew, the Petri-net implementation started limiting the models development. Expansions to Timed-Coloured Petri-nets expanded the flexibility, However problems related to the synchronisation of tokens and transitions, in that transitions can only fire if every place connected to the transition contains a token, became more pronounced as the asynchronicity of agent communications was simulated. Additionally while transitions have guard functions, places do not, which led to 'wasted' transitions that were only employed to provide the guard functionality. Realistically the largest problem was that the Petri-nets had to be created programmatically in AnyLogic with each permutation of the agents development.

5.4.3 Statecharts

Following on from these limitations, Statecharts another widely used activity-based modelling technique, (Harel, 1987) were then chosen to replace Petri-nets as the control structure. The definition of a statechart (from Eshuis (2005)) is a tuple $(N, H, source, target, children, type, n_i)$ where:

- N is a set of nodes,
- H is a set of hyperedges, $N \cap H = \varnothing$,
- $source : H \rightarrow N$is a function defining the set of input nodes for each hyperedge,
- $target : H \rightarrow N$ is a function defining the set of output nodes for each hyperedge,
- $children : N \in N$ is a function that defines for each node its immediate subnodes. If a node has children, we call it composite.
- $type : N \rightarrow BASIC, AND, OR$ is the function defining the type for each node. Each non-composite node is BASIC. Each composite node is either AND or OR. If an AND node is active, all its children are active as well. If an OR node is active, one of its children is active. (So OR is actually a XOR.)
- $n_i \in N$ is the initial node.

There are many similarities between the two activity based modelling languages, for example Places are similar to Nodes, Transitions to Hyperedges, and the source and target functions correspond to the flow relations. Rik Eshuis (2009) provides a comprehensive formal translation from safe Petri-nets to statecharts. The author therefore used this mapping and it was valid at the time, because the petri-nets did not contain any of the extensions that could have caused the mappings to fail. Figure 5.3 shows the equivalent statechart to the petri-net in Figure 5.2

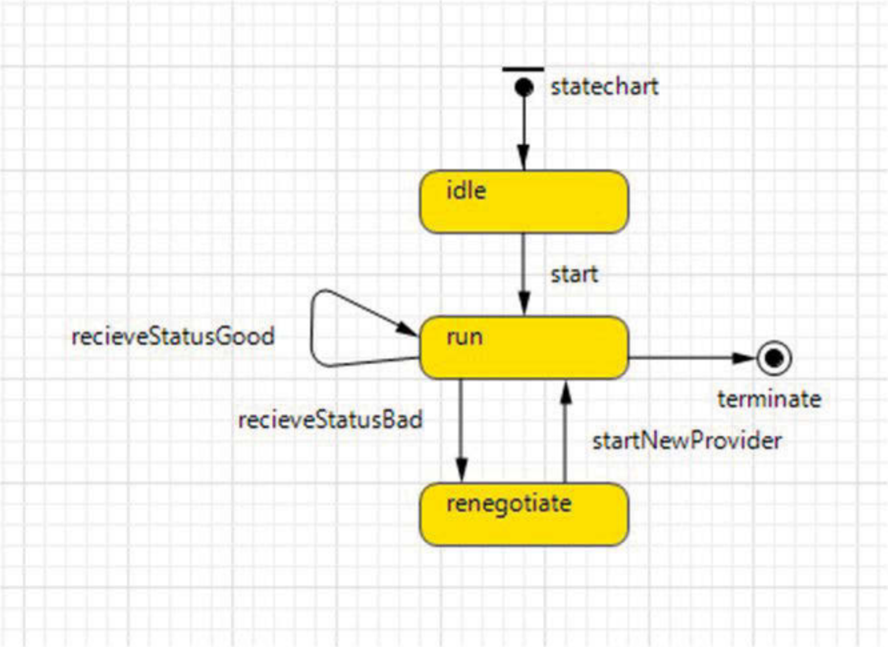


Figure 5.3: Equivalent Statechart to the Petri-net of the BSA bottom control. 5.2

As the action cycles continued the state-based model was expanded to simulate the negotiation and provisioning of bundled services in an open market. The final model follows the architecture described in Section 3.5 and contains Bundled Service Agents (BSA), Single Service Agents (SSA), Service Service Resource Agents (SSR) and User

Agents across a market environment. For the reasons discussed in Section 5.1 the model is a high level view of the system, focussing on the agents influence on service provisioning, and management, specifically dealing with resilience and recovery

5.4.4 Bundled Service Provider and Agents in the Simulation

Following the architecture in Chapter 3 (page 33), and the Design of Chapter 4 (page 52), the final implementation of the simulation has simulated as much of the functionality of the BSP as possible in the environment. The control structure of the BSP and its agents has been implemented in Anylogic statecharts, with some specific omissions, related to the unique business decisions such entities would be required to make. Such business decisions examples include specific service type focuses and restrictions, such as choosing to manage and provide only requests that come from a certain geographical area, or choose services from only some Single Service Providers. Additionally In the final simulation implementation all BSPs quote on all contracts as this encouraged more competition in the limited simulation run. The state machine described in Section 4.2.2 was implemented across three layers:

- Figure 5.4 – The Bundled Service Provider (BSP)
- Figure 5.5 – The Bundled Service Agent (BSA)
- Figure 5.6 – The BSA Single Service Control

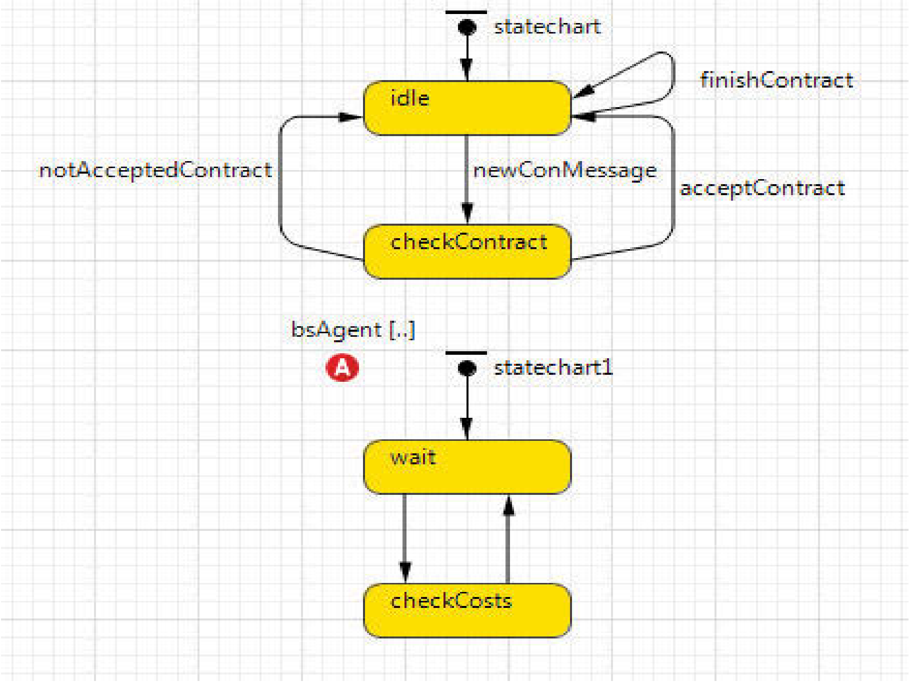


Figure 5.4: State Machine in the Simulation of the BSP control.

Figure 5.4 shows the state machine for the provider layer of the BSP. The BSP receives the contract, and creates a new BSA to provide a quotation and manage the service if the quotation is accepted. Figure 5.5 shows the upper level BSA behaviour, at this level the BSA is responsible for assembling the optimal single service providers for the contract and retrieving quotes from the SSP's. Figure 5.6 shows the implementation of the BSA's single service control, including the failure related decision structure described in Section 4.2.2.

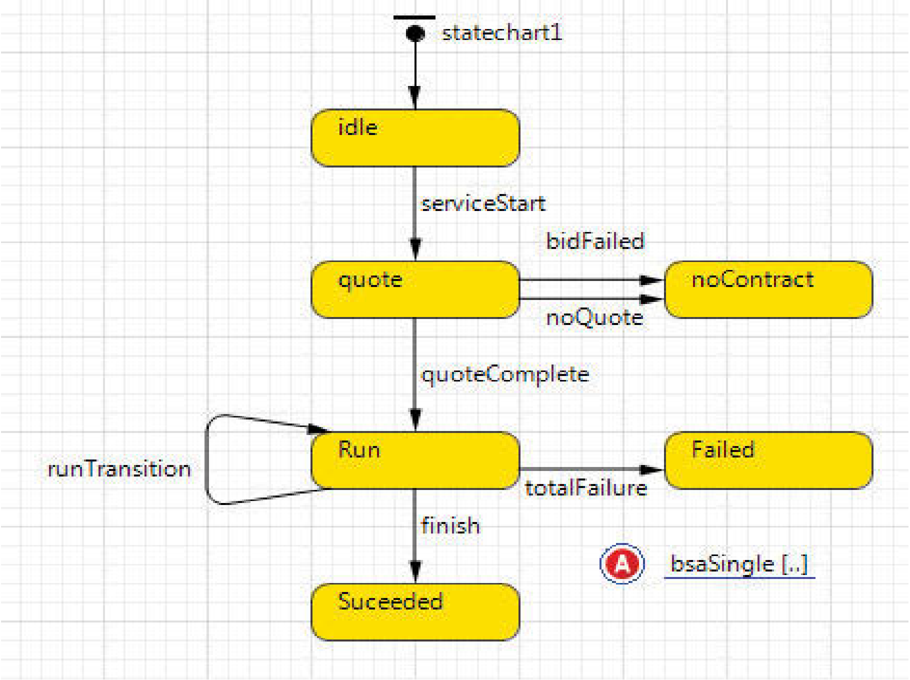


Figure 5.5: State Machine in the simulation for the BSA control.

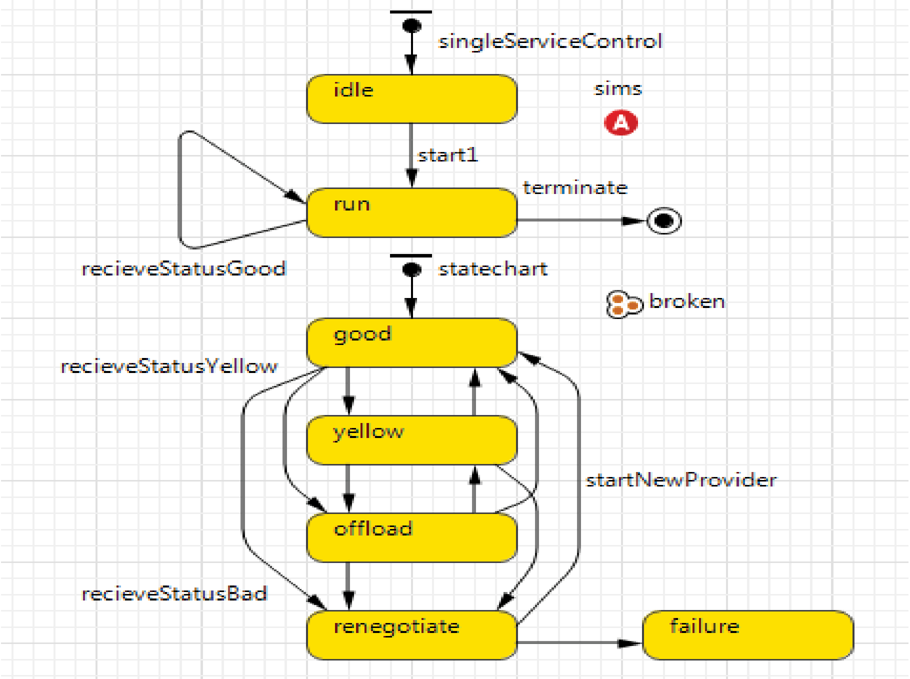


Figure 5.6: State machine in the simulation for the BSA single Service Control.

As it is assumed that other than instructing service providers of the new service provider on re-provisioning, the BSP would be uninvolved in the mechanics of service connection.

Reconnection ability between various providers on receipt of the contract specification is assumed.

Before describing the mechanics of simulating the BSP's risk management through the SIMS, the implementation of the other entities which would participate in CARMAs marketplace are described. Again for these entities as with the BSP and its agents, the entities have been implemented at a high level, with a focus on their interaction in the marketplace and not on their individual service and management characteristics.

5.4.5 User Agents in the Simulation

The User Agent builds contract requests utilising the high level information model sections described in Section 3.3, and submits them to the BSPs in the market. They further are responsible for generating the usage to the SSPs and SSRs. Figure 5.7 shows the two state machines responsible for the described behaviour. ServiceStart, generates the usage, and contractNeg chooses the cheapest quote.

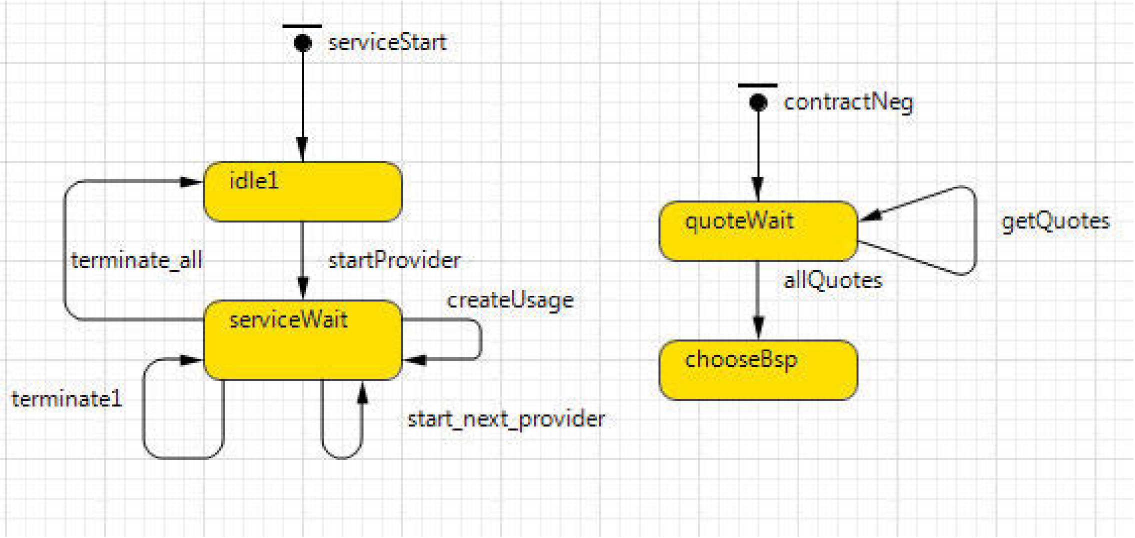


Figure 5.7: The state machine for the User Agent.

Again the mechanics of connection, other than being specified abstractly in the contract, are omitted.

5.4.6 Single Service Providers and Agents in the Simulation

The Single Service Providers and their agents have been simulated to send internal monitoring messages on the current services performance. This allows the model to simulate the basic functionality of re-provisioning failed services with another provider for the remainder of the contracted time. Figure 5.8 shows the state machine for the SSA. Once the service is started, shown by the serviceStart state machine the SSA interacts with the BSA via status messages, which inform the BSA of the SSR status, shown in the checkStatus state machine. The information gathered for the status messages is related to the current utilisation of the SSR as well as the delay on the usage messages sent through the simulation.

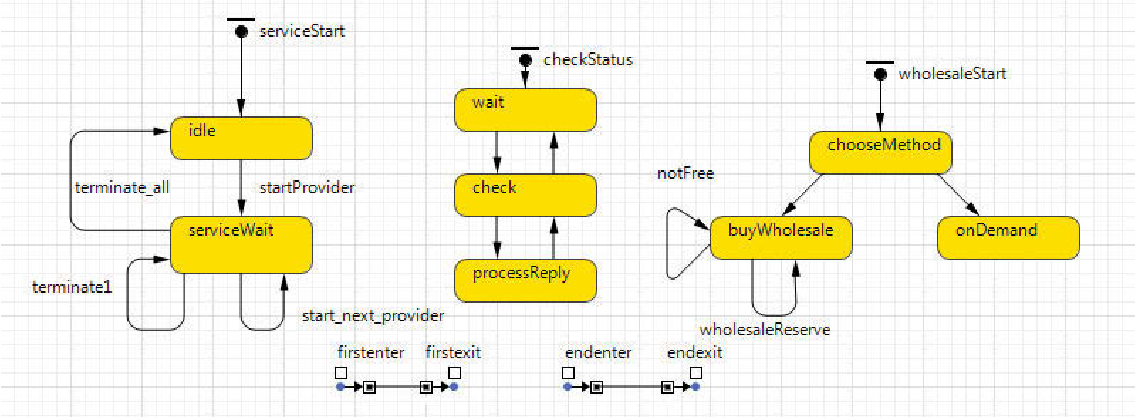


Figure 5.8: The state machine for the Single Service Provider.

Each SSA in the model has access to one Single Service Resource(SSR), and, is responsible for provisioning and scheduling the resource across the SSR. Additionally, the SSP chooses whether or not to pre-purchase resources on the SSR or to purchase them on demand.

5.4.7 Single Service Resources in the Simulation

All Single Service Resources (SSR) are modelled as abstract resources, divided into four components of differing QoS. Usage Messages that are created by the User Agent are passed through the paths labelled, as High, Medium, Low, and Best Effort and the paths have appropriate queues sizes for network connectivity and virtual machine/application processing, with delays relating to their individual priorities.The messages themselves are simple representations of either packets, with simple header information for routing. Even in more complex services, such as video conferencing, streaming services, or cloud based distributed services, the fundamental elements are packets of varying sizes and header information. The delay is a Rayleigh distribution based around a percentage of the model time (10%). Rayleigh distribution was chosen as it reflects the variability of traffic, from the point of view of bit rates in the case of connectivity or response times in the case of cloud computing IaaS. A resource pool for resource units represents the percentage of utilisation in the cloud or network infrastructure, in the simulation this is currently also set to 10%. Greater utilisation of the Cloud/Network resource results in poorer overall performance affecting lower quality services first due to the priorities set on the resource pool with regards to the quality path. In the simulation the maximum number of contracts that can be provisioned without affecting performance is dependant on the message rate, the delay distribution and the size of the resource pool. Figure 5.9 shows the structure of the resource. It should be noted that the best effort queue shares in the resource pool though it is not explicitly shown in the diagram.

The SSR also contains a reservation system. The reservation system restricts the reserva-

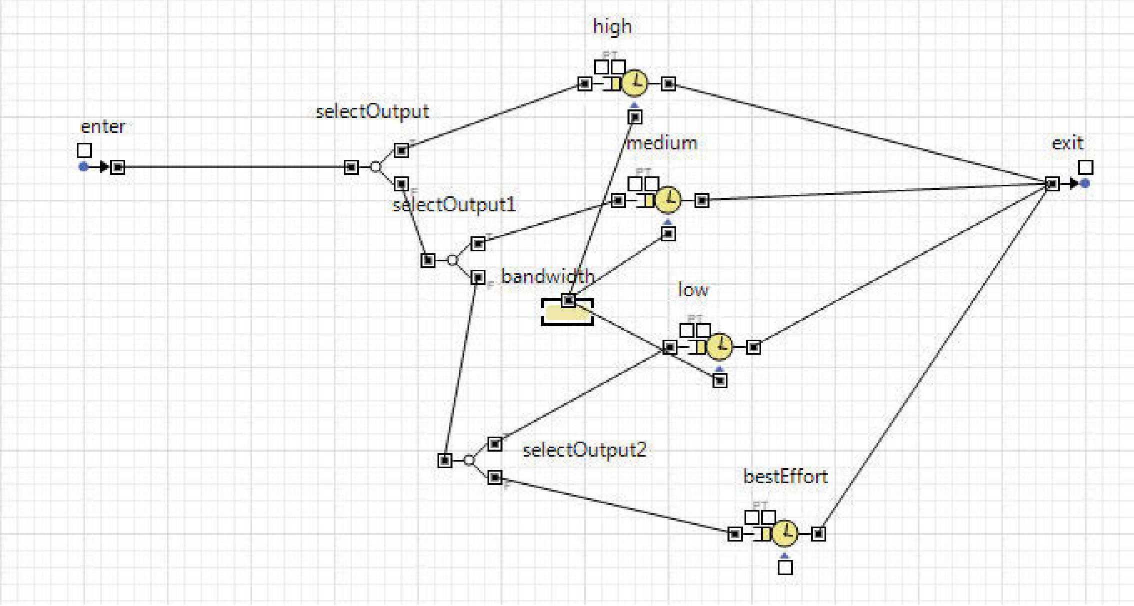


Figure 5.9: The queuing model for the Single Service Resource.

tion of SSRs by the contracts to a maximum of four concurrent providers. The SSR can be assigned to one or more SSPs.

5.4.7.1 Service Coverage in the Simulation

One further point with regards to the simulation of the service types covered in Section 3.3 in the SSRs is that the service types are deliberately represented in abstract terms in the simulation. In practice the sheer number of individual services that could be developed in a global platform such as CARMA makes the specification of individual limitations problematic. In general each individual service type would contain unique metrics that would be measured with a focus on the particular performance characteristics of the service. Though while the metrics are unique the individual characteristics do conform to some generalities which are mentioned above in Section 5.4.7 such as queue size and resource pool size. The specification of these performance characteristics in CARMA however, would be the responsibility of the individual SSP's and SSR's, collated into the previously defined four colour codes of performance status, described in Section 4.2.2. This assumption of collation allows the simulation to treat each service type as homogeneous entities. Treating them as such, allows the simulation to increase the number of services simulated as well as allowing a focus on service interaction without increasing the simulation complexity needlessly.

5.4.8 Service Information Management System in the Simulation

The functionality of the SIMS is driven by the two identified goals shown in Section 4.4: judging the risk of contracting a new bundled service and judging the risk of continuing, or failing over the service under poor performance criteria. This functionality is split across different parts of the BSA, with the functionality that judges the risk of accepting a new contract being located in the BSA, and the functionality of judging the risk of failing over to a new provider displaying poor performance is located in the BSAs Single Service Control modules.

In the simulation, the BBN of the SIMS is implemented using 'unbbayes' (Costa et al., 2008) which is an open sourced Java based application for modelling probabilistic networks. The unbbayes modelling application was imported into the simulation and set up using the Bayesian Belief Network shown in Section 4.4. A database was also created to store the values for the Historical Service Performance node for the different Single Service Providers as seen by the BSPs.

Following the design, all Single Service Providers send messages that correspond to four simple states, that being either green, yellow, orange or red, which indicates the service is fine, having low level problems, having high level problems and failure. Due to the degree limitations of the messages that are being sent from the Single Service Providers to the BSPs it was decided to correspond those messages with the states in the Historical Service Performance node, with the number of green messages corresponding to the percentage of green messages received, the number of yellow corresponding to the intermittent percentage, the number of orange with high latency percentage and the number of red corresponding to failure.

In reality the message information would be determined by the contract between the BSP and the SSP and the required information would vary dependant on the service that the SSP was selling.

The results of the Market Action node of the dBBN are combined with the cost of the provider using the formula:

$$\text{Cost} \times \lg((1 - \text{Risk}) \times \text{Scaling Factor}) \tag{5.1}$$

The probability of the 'provide quotes' outcome in the Market Action node was used as the risk probability in the formula. As the range of the percentage output of the dBBN is relatively small, this risk is rounded to 3 decimal places and multiplied by a scaling factor of 1000 before the log of the risk is taken to exponentially expand the difference in risk judgement. This formula is used to assign an edge weight to a Single Service Provider weighted directed graph. Since the late 70's there has been an understanding of the natural similarities between random graphs and computer networks (Newman, 2005). Graph theory has been used extensively for the optimisation of network paths. As such it forms an important part in the decision making process for the choice of Single Service Providers in a bundled service. Dijkstra's algorithm (1959) is then used to calculate the optimum path for the bundled service based on the weightings.

Initially it was decided that the historical service performance figures were to be calculated per Single Service Provider, rather than per service quality level. The reasoning is that all providers would wish to compete at all levels of quality that could be decided by the contract specification, and that the risk of contracting with a particular provider is independent of the quality of service that is required. Further, the simulation of the SSR was built to represent current management systems that employ quality of service measures, which take steps to ensure that under congestion scenarios, lower priority or lower quality services receive lower resources, resulting in higher failure.

However, upon analysis it was observed that the employment of priority queuing by the SSRs adversely affected the ability of the BBN to judge the risk of contracting services with particular providers as the failure of a low quality service at a provider resulted in the BBN returning a poor judgement for the provider when presented with a high quality service. This meant that the BSA would choose a less optimum path with resulting higher costs. This is explained in greater detail in Section 6.4.2.1.

In all cases the BSP wishes to contract the service with the provider with the lowest risk of failure, even if they have chosen a lower quality of service and are accepting of greater risk than a high quality service.

The second sub goal of the resilience goal is to judge the risk of contracting the new provider when the old provider is failing. Due to the limitations of the simulation, it was decided to again split this sub goal again into two parts. The first part being that if another provider exists at the same node as the failing provider, then the BSP should switch to that provider, as changing to this provider is deemed less of a risk to the whole provider than determining a whole new path, regardless of the individual risk inherent in the new provider. If the other providers in the node are not available, then the BSA has to make the drastic step of building a new path around the failing node. To do this the BSA again uses the same method as for the first goal, that of calculating the edge weights, using the dBBN and then optimising via Dijkstra's algorithm.

5.4.9 Contract specification and negotiation in the Simulation

In the model the exact specifics of the resources needed to provide the services are left out, for example the specific router or server locations and specifications. It is assumed that such detail would be provided by the specific providers management systems if in negotiation with the infrastructure providers.

There will always be a slight delay in negotiating contracts, resulting in a slight delay for on demand services and service discontinuity when dealing with service changes. However, this will be much less than the delay involved in manual reconfiguration.

Finally, while the graph transformation and specification utilising the TMForums SID was initially added to the system, the additions slowed the simulation run times considerably (due to the simulation environments limited resources), and since the service resources were also generalised it was felt that such additions were not required for validation. Instead the contracts were kept at an abstract level in keeping with the requirements stated in Section 3.3.

5.4.10 Other exclusions from the simulation

Overall, as previously stated, the Simulation omits the Single Service Providers and Single Service Resources mechanics of providing the connections. It is assumed that such information is provided in the negotiated contracts, as specified in Section 4.3.2 and that the responsibility for fulfilling such requirements is the responsibility of the SSP's and SSR's own management systems.

With regards to security, the model does not take into consideration the concerns of ensuring data security with the various service providers it engages, other than to specify the types of service (such as a VPN).

With regards to the marketplace, the simulation employs only the primary request for quotation market, with the implementation of the secondary market left as future work. Similarly, while the wholesale fixed price network is employed, examination of the profitability of such a market was not deemed relevant to the validation of the system.

5.5 Simulation Study Design and Evaluation

The simulation previously described in this chapter is a model of the CARMA design including an abstraction of lower level network management systems and their interactions, shown in Section 5.4.7, and created from knowledge of those systems. An industrial case study model which uses different types of bundled services was also created. The application of the case study model to the CARMA simulation was used to design the type and nature of the traffic which would be generated to provide overall evaluation of CARMA via simulation. To ensure the validity of the outcomes of the case study, it was constructed to conform with the four quality considerations devised by Yin (2003).

5.5.1 Validity of the Case Study

Yin (2003) proposes four properties in his work on evaluating case studies for effectiveness. These properties Construct Validity, Internal Validity, External Validity, and Reliability were used to define the effectiveness of overall evaluation of CARMA via simulation. The definitions of the properties are directly below, followed by the application of those definitions to the CARMA simulation evaluation.

Construct Validity Refers to the conformance of the measured results to the stated goal of this case study, which is to establish the effectiveness of CARMA's ability to provide responsibility in a multi-provider environment consisting of cloud computing and telecommunication services. To achieve this, each phase of the simulation results was evaluated against the requirements for profitability, reliability and resilience.

Internal Validity Focuses on the causal relationships between dependent and independent variables. In this case the independent variables are the number of providers in a contract, the performance of the individual providers, and the quality of the required service. The dependent variables are the reliability and profitability of the Single Service Providers, over the course of the simulation run. The expectation is that due to the influence of CARMA, that the profitability and reliability of services would improve over time.

External Validity Is concerned with the domain of the case study beyond the context of the immediate contract requirements. Based on the requirements of CARMA, the particulars of the case study, such as the telecommunication and cloud service providers can be generalised. As such, the case study is independent of the individual telecommunication and Cloud providers.

Reliability Addresses the possibility of another person applying the proposed methodology to a different problem in a different organisation. This reliability is attained through the applicability of the CARMA case study to any usage of telecommunication and cloud provider services.

The case study is based on a on-line travel company. The on-line travel company's website contains information and bookings on tours, accommodation, attractions and travel, alone or in any combination. Overall the sheer size of the content available in this travel agency makes accessing the data from a central location result in poor performance with high latency and poor usability. In order to improve on the performance of the site the company wishes to split the functional logic of the site from the advertising and content, placing the content with different geographical cloud providers, and leaving the site logic in one central location. In order for this type of setup to be effective in the past, websites used static geographical locations, paying content delivery companies a flat fee subscription for use of their content data centres. The amount of content served and the bandwidth available to a particular website is statically pre-arranged and generally unchangeable without significant, undesirable, lead time. Further the connectivity between the central functional website and the content servers is almost always best effort, resulting in little or no control over the different application sections interaction.

5.5.2 Bundled Service Market effect on the Case Study

With the use of the Bundled Service Agent, the three services of website logic, content server and the network connectivity could be dynamically served in periods of high usage in particular regions, with content servers, represented by remote virtual machines, placed in different cloud environments based entirely on their particular geographical locations as highlighted by Figure 5.10. As the bundle includes QoS guaranteed tunnels between the various content servers and the central location, latency caused by poor connectivity or high network utilisation would be avoided. Additionally, as the content servers are only set up in locations as they are needed, there would be certainly cost benefits over the traditional flat subscription rates. Further as performance information about the service providers is stored by the Bundled Service Agents SIMS, the Bundled Service

Figure 5.10: Graphical representation of the case study example, showing one user utilising multiple providers in different geographical locations.

Provider can take responsibility for the performance of the application, through initially simple methods of choosing the most reliable providers, and reserving and offloading on secondary providers in the face of poor performance.

5.5.3 Extension of the Coverage of the Case Study for Simulation

This case study, although based in an industry example and thus with an inherent validity is, in terms of testing the capability of CARMA, limited. The designed environment for the CARMA system allows for multiple simultaneous contracts interacting in the dynamic market place. To extend the case study for simulation the case study needs to be modified to cover the expanded simultaneous contract and service requirements. To do so the particulars of the case study, such as a virtual machine for the content servers in different geographical regions, was abstracted in the contract generation keeping the important aspects of VMs in different geographical regions but discarding the specifics of the VM use as the simulation does not want to be limited to just content server VM use. Likewise the guaranteed QoS tunnels were represented by connectivity contracts with individual providers in the simulation on a path between some initial location and the contracted VM. These basic contracts were then replicated numerous times through the simulation lifetime with randomly generated contract lengths, quality requirements and endpoints. In the simulation the contract lengths were all specified with start and end times, but were generated with minimal lead time, representing the 'at need' requirement of content servers in the case study. For quality requirements the contracts were specified at the abstractions of high, medium and low only with the best-effort contracts removed from the simulation as the requirement for paying premiums on contracts which do not utilise any QoS would be unlikely. With regards to the endpoints of the contract, it was decided that the re-provisioning of failing providers at these endpoint nodes would only occur with other providers at the node. This was done to reflect the current technical ability of public cloud providers. For example, Amazon has the ability to migrate services between availability zones in a particular region. However, as yet, there is no proven ability to live migrate between regions (Section 2.7).

5.6 Summary

This chapter has presented the structure of the simulation built to model the CARMA design. The high level nature of CARMA, being designed to interact with multiple providers across a market environment, above the level of individual telecommunication and cloud service management systems meant that the traditional low level approach to network simulation was inappropriate. Rather CARMA was simulated in a more abstract manner. One in which the interaction of the entities was deemed more important than the restrictions of the individual service device provisioning.

The CARMA simulation simulated the interaction of the Bundled Service Providers, its agents and its agent's single service controls, as well as a combined Single Service Provider and agent and a system dynamic based service resource and agent. As the purpose of the simulation was to determine the viability of the system with regards to risk management and resilience, the lower level mechanics of connection, between users and the individual service providers as well as the mechanics of connection between the service providers and the resource providers was assumed and omitted, and the exact contract transformation and specification, detailed in the last chapter (Section 4.3.2) was also generalised and maintained at the highest level of abstraction.

While these omitted aspects of the design could be seen as limiting the validity of the simulation, initial and subsequent reflection on the simulation design, especially in regards to the validity of risk management, did not require the simulation of the network connectivity protocols or specific values in contract evaluation. One of the stated goals of the CARMA design was to introduce risk management in an environment of heterogeneous implementations and business entities. This meant that the implementation of the risk management had to be independent and unaware of the particular network characteristics of individual Single Service Providers and their Single Service Resource Providers. Characteristics like device specifics and manafacturer limitations.

The Bayesien Belief Network was designed to be identical for all BSPs in the simulation with only one method of implementation. While this certainly wouldn't be the case in actual competing BSPs where the design of the Bayesien Belief Network would offer a competitive advantage to the individual BSPs in the marketplace, identical implementations in the simulation allowed the validity of the basic approach to risk management to be more completely verified.

To assist in the validation a simulation based case study was developed based on the four properties for case study effectiveness, as an example of the type system on which CARMA would be employed.

This case study forms the basis of the contracts and initial simulation parameters against which the simulation was run and the results analysed. The next chapter shows the results of the simulation runs and includes the analysis of these results. Included in the analysis is the changes and refinements of the initial system design that were the result of the analysis and reflections on the analysis. The reflections also form the basis for the discussion of the assumptions and issues discovered through the simulation.

Chapter 6

Results and Discussion

The preceding chapters have outlined the simulation design which was predicated on the requirements of Chapter 3 and the CARMA design outlined in Chapter 4. To verify and validate the design of CARMA the simulation was repeatedly run, analysed and evaluated against the goals of this thesis. This chapter presents the results of this analysis and evaluation and a discussion of the issues raised and assumptions challenged, as well as the refinements the discussion led to. As such the results are presented in a somewhat chronological order, with discussion and reflection sections where appropriate.

6.1 Simulation Detail

Initially, the simulation environment was set up to involve multiple randomly generated contracts, which were derived from the case study described in Section 5.5.2. They involved a minimum of two service types, and a minimum of two required service providers. The start and end times were always set and were randomly generated. The simulation environment created 26 nodes of Single Service Providers, labelled for typical geographic locations as shown in Figure 6.1 where the Single Service Providers, representing the telecommunication and cloud providers, have been placed in key points of a map of international fibre channels (Johnson, 2008). For the majority of the simulation runs (unless specified) the environment created two providers per node and assigned a resource (SSR) to each provider (SSP). To introduce poor reliability for the simulation half of the SSRs received a smaller resource pool for load management. Table 6.1 shows these initial simulation parameters for the Single Service Providers.

The simulation was then run for a simulated time of one year with the simulation incrementing in one minute intervals, which resulted in the generation of around 26,000 accepted contracts or a rate of 1 every 20 minutes. A new user was generated for every contract and each contract utilised a differing number of resources that was dependant on minimum resources required to connect the endpoints specified in the contract. This minimum number of resources was determined by the SIMS utilising Dijkstras algorithm

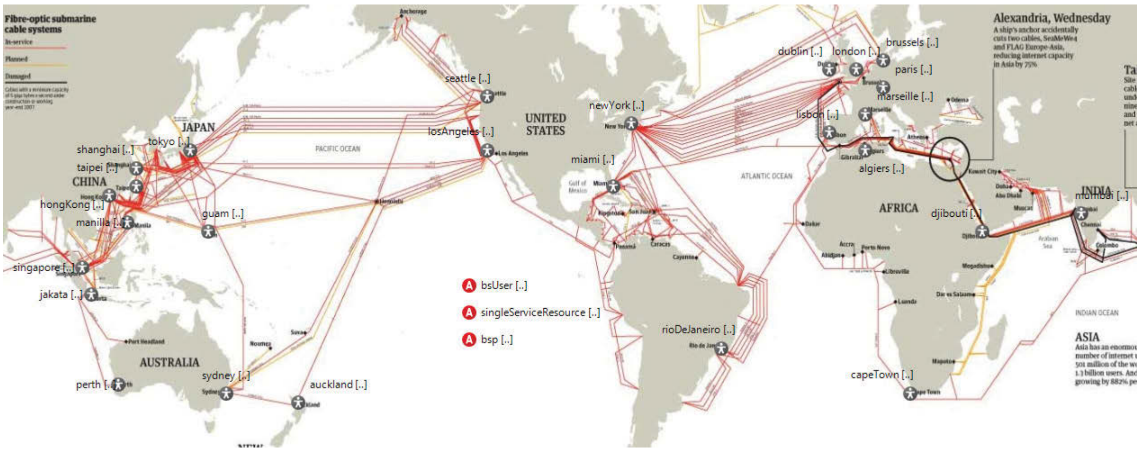


Figure 6.1: The Map of Single Service Providers in the Simulation, Image taken from (Johnson, 2008).

as specified in Section 5.4.8. Each endpoint utilised one of the six defined service types connected by connectivity resources. The users of the contracts then competed for the resources as specified by the revised generated contract created by the BSP's. While Best Effort contracts have been included in the design, they have been excluded from the simulation as in such contracts, the performance risk judgement and resilience controls would not be required.

Provider	**Reliability**	**Pool**	**Quality**	**Cost Range Per Minute**	**Priority**
Provider 0	Unreliable	80	Low	0.01 ... 0.039	Lowest
			Medium	0.04 ... 0.069	Middle
			High	0.09 ... 0.119	Highest
Provider 1	Reliable	100	Low	0.02 ... 0.049	Lowest
			Medium	0.05 ... 0.079	Middle
			High	0.10 ... 0.129	Highest

Table 6.1: The initial system parameters of the Single Service Providers in the simulation.

6.2 Overall Goals of the System to be Validated

The aim of the CARMA system to date is to provide responsibility for end-to end services, where responsibility is defined as the judgement of risk, the reliability of the providers and the provision of resilience. Specifically these goals are:

1. Responsibility through risk calculation, resilience.
2. Guaranteed QoS
3. Increased utilisation for Single Service Providers and Service Resource Providers
4. Decreased Price

6.3 Verification

The simulation has to conform to the design parameters and behaviours that were described in Chapter 4. To verify this behaviour the simulation has to perform the following functions:

- The Bundled Service Agent takes appropriate steps in the case of poor service performance, up to and including recovery from failure,
- The monitoring messages sent are appropriate to the threshold values assigned,
- The performance of the Single Service Providers move towards a steady state, and
- The QoS for the individual services is reasonably simulated.

6.3.1 Service Recovery

The purpose of this examination is to confirm that the appropriate action, renegotiation of failure, was undertaken on the determination of poor performance of the initially contracted provider. For service recovery the BSA has three options, the first being a full fail-over, moving through the four states described in Section 4.2.2. The second being the move through reserve, and offload, but on receipt of better performance messages the service cancels offload. The final option is that the service lowers quality requirements in cases where re-provisioning is impractical. The graphs in Figures 6.2 and 6.3 displays the utilisation of the Single Service Providers (load and queue size vs time) by the Bundled Service Agent's single service control.

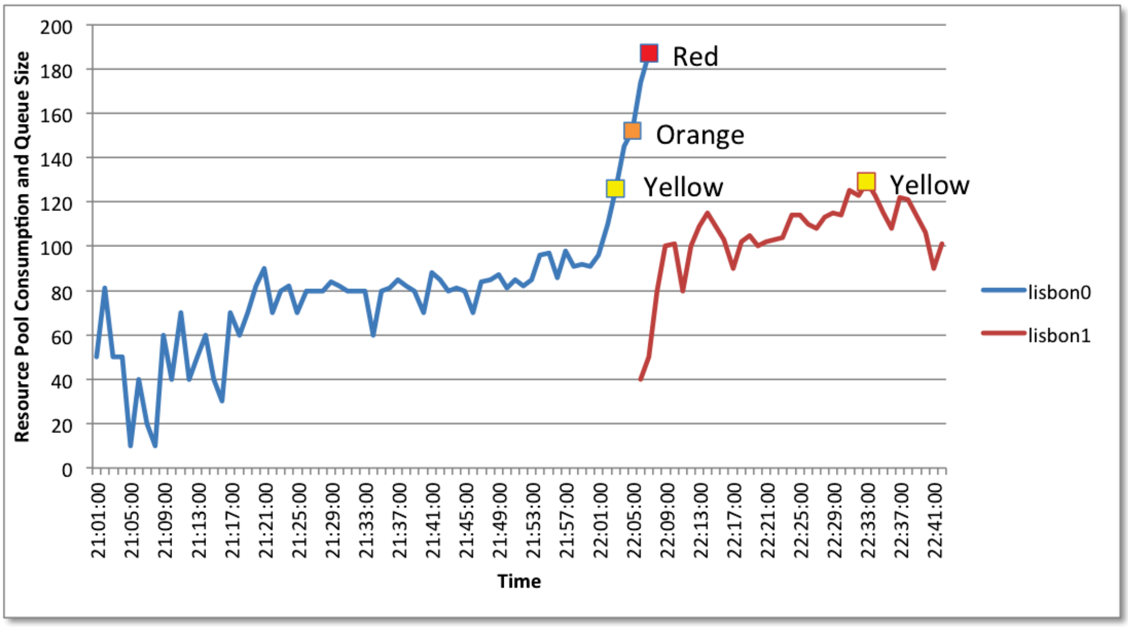


Figure 6.2: Service fails over from one provider to another on the same node.

Graph 6.2 shows the that the service provider experiences problems at just past 10pm, with the BSA's SIMS starting a reservation with the new provider, indicated by the yellow marker. At the orange marker the BSA's single service control starts offloading to the new provider, and when the service fails (indicated by the red point) the secondary provider (lisbon1) retains full control.

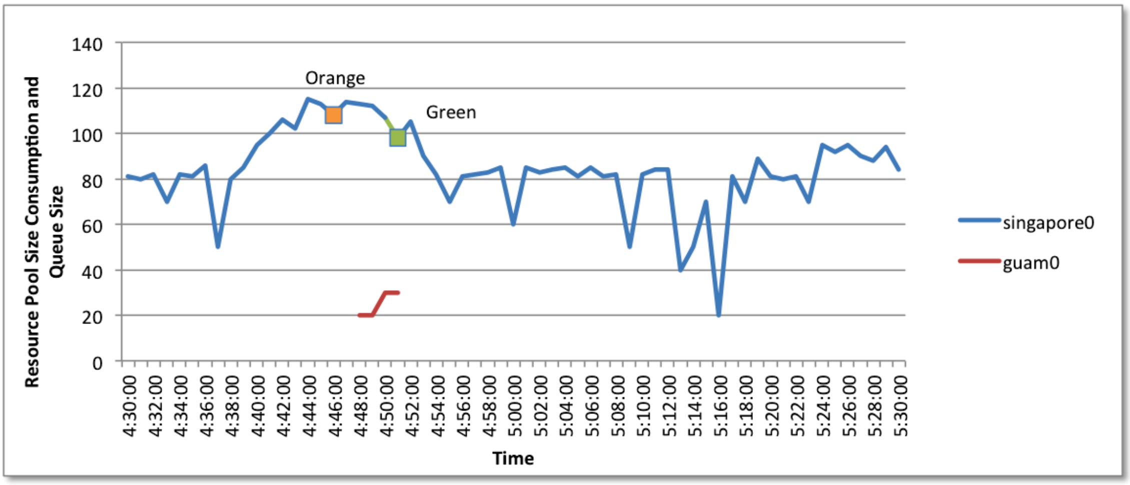


Figure 6.3: Service Offloading but no failover.

The graph in Figure 6.3 shows the situation when the BSA' contracts a new provider, starts the offload, but then has the initial service recover. In this case the BSA cancels the offload to the secondary provider and continues the service with the primarily contracted provider.

6.3.2 Monitoring Messages are correct based on the threshold values

To verify that the monitoring messages sent from SSA's to BSA's were based on queue threshold values a selection of messages sent from a single SSA to the BSAs were examined. For this SSA the threshold values were set at 20 items in the queue for green, 60 for yellow 100 for orange and anything over 100 was set to red.

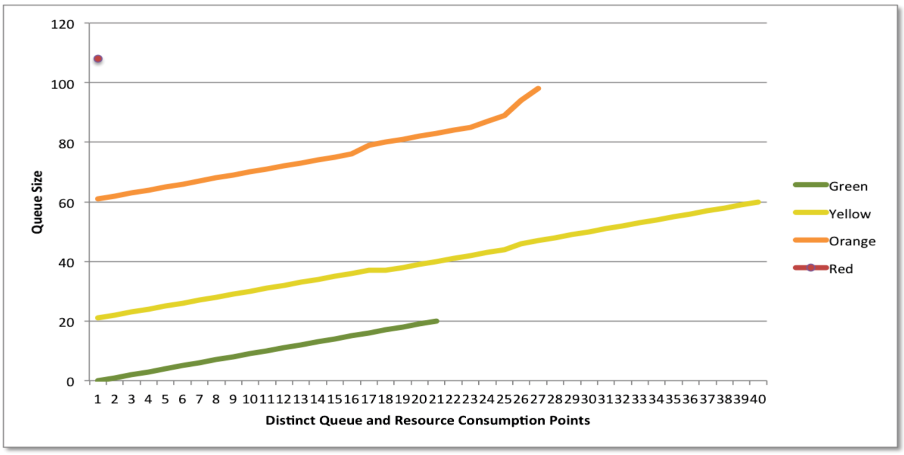


Figure 6.4: A selection of messages with colour assignation plotted against the queue size.

Figure 6.4 shows that the appropriate message was sent for all values sampled.

6.3.3 Simulation Runs to a Steady State

To verify that the performance of the BSP's agents move towards a steady state as the simulation runs, the messages received by the BSP's were examined over time. Specifically the number of Green messages over the number of Red messages were examined per week.

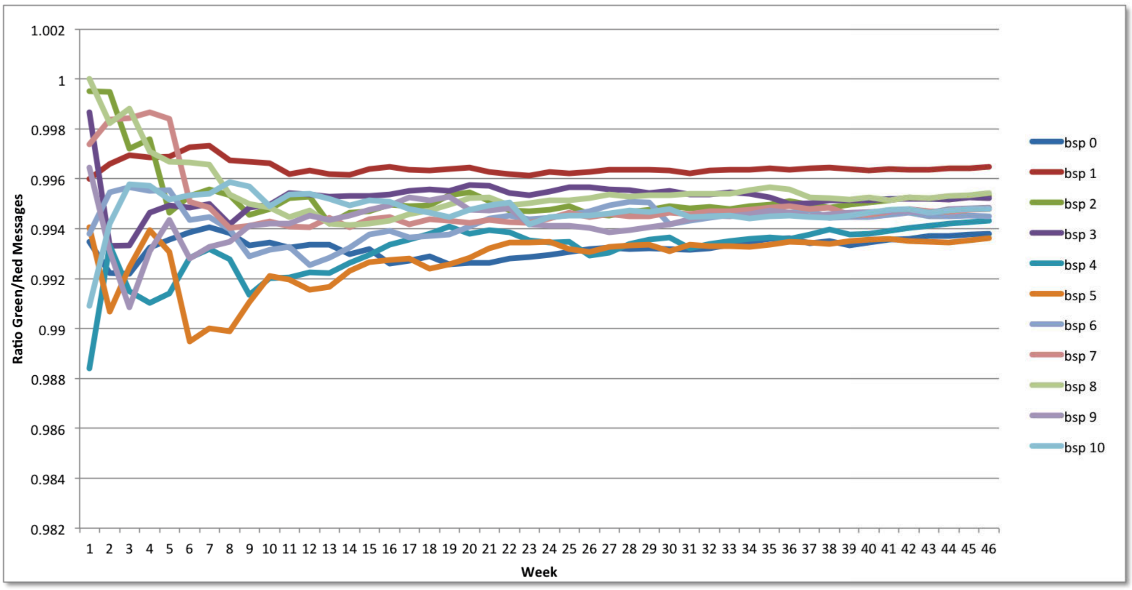


Figure 6.5: Ratio Of Green to All messages for the first 10 providers.

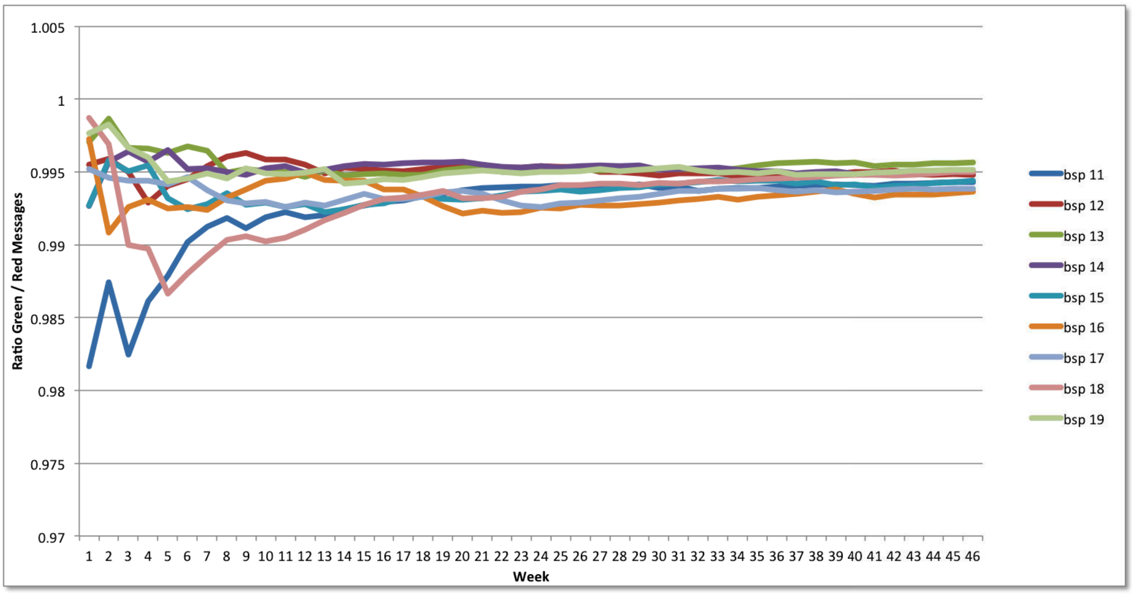


Figure 6.6: Ratio Of Green to All messages for the next 10 providers.

The results show that the simulation performance does move towards a steady state around the probability of failure of the SSR. This probability is based on the Rayleigh distribution of usage message rates and the number of providers that can simultaneously exist at a SSR before failure.

6.3.4 Reasonable Simulation of QoS behaviour

To determine that the Quality of Service properties of the services simulated through the resources conforms to expected behaviour of telecommunication systems, the simulation's message traffic was examined with regards to the status messages received by each Bundled Service Provider for provided services at each level of quality. For clarity these quality settings were High, Medium and Low and the message classifications were green, indicating no problems, yellow, indicating some problems, orange indicating large problems, and red, indicating failure.

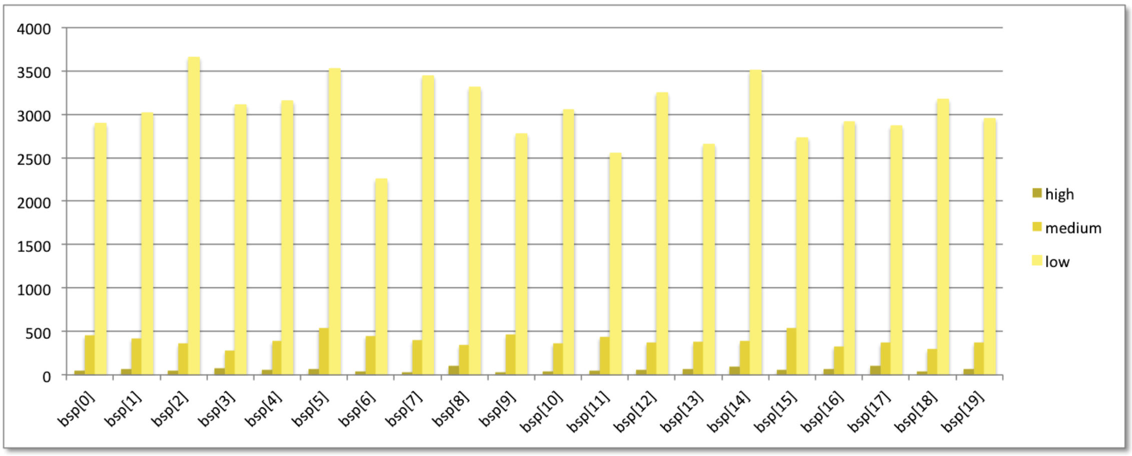


Figure 6.7: Yellow messages at High, Medium, Low quality settings.

The results are as expected with the number of yellow messages for Low quality services being significantly higher than Medium, which is also higher than the High services.

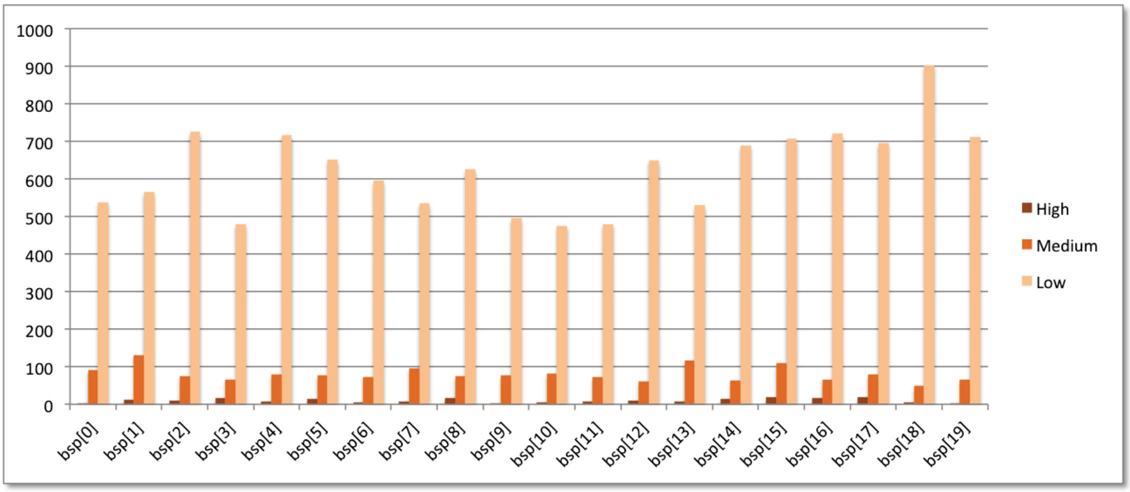


Figure 6.8: Orange messages at High, Medium, Low quality settings.

Further the orange messages show the same distribution. With orange messages being much greater for Low services than Medium or High.

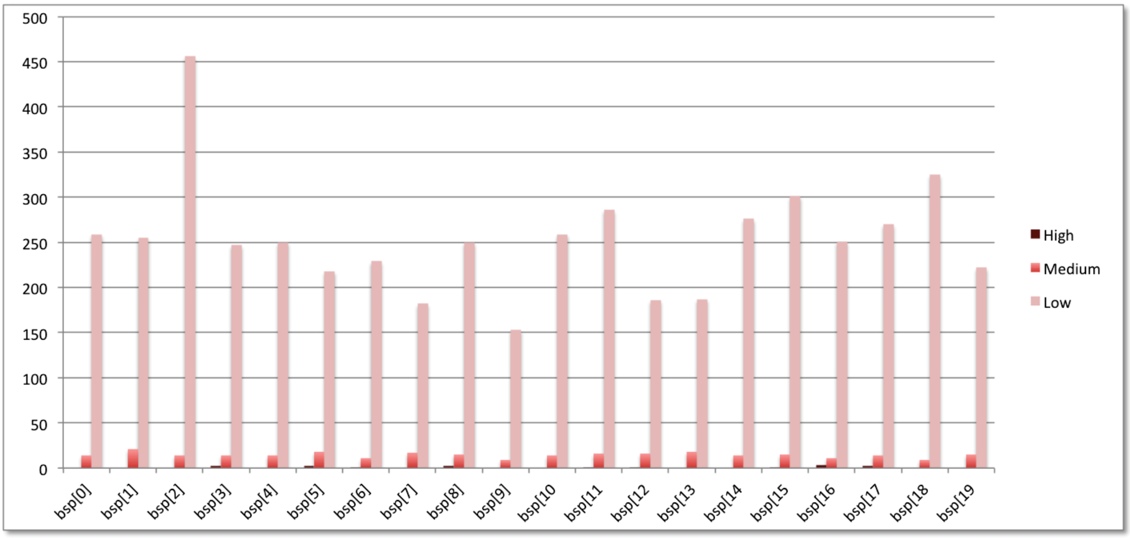


Figure 6.9: Red messages at High, Medium, Low quality settings.

Finally for the red messages the same pattern holds, with Low quality services experiencing a significantly larger failure rate than Medium services, and High quality services experiencing barely any failure at all.

Also shown in these three graphs, is that the overall number of problem messages decreases with the severity of the message. Taking Low quality services for example, the graphs show that there is significantly more yellow messages, which indicate small service problems, than red messages, which indicate failure.

6.4 Validation

To validate the model, the simulation needs to analysed with regards to the stated aims of this research, namely responsibility for end-to-end services, increased utilisation for Single Service Providers, and scalability.

The structure of this validation follows action research cycles that focussed on each particular aspect of the stated aims. For each cycle a simulation run is planned and executed, with the results being analysed, observed and reflected on. The reflections and decisions made affect the planning of the next cycle. The sections are named for each validation goal.

For the stated aim of Responsibility, there are two goals, Risk management and Resilience.

The first cycle focuses on the goal of resilience, which at a basic level is demonstrated by Section 6.3.1 with the models ability to recover services under failure. However, there are further questions that needed to be explored such as, with this verified service recovery, do any services fail? and if so, for what reason? How much does this recovery affect the profitability of the BSP?

The second and third and forth cycles focuses on the management of Risk, as the model needs to demonstrate that each of the BSAs do, over time, choose the most appropriate SSPs for each service. In this case the most appropriate is determined by reliability and cost. Further, the model needs to demonstrate that the use of risk management improves the reliability of the services. Finally, the simulation needs to demonstrate that the use of risk management increases the profitability for the BSP.

The fifth and final cycle focuses on the goal of increased utilisation, where the simulation needs to demonstrate that under the same service conditions the use of CARMA, with risk management and the focus on minimal cost in the market environment, increases the overall utilisation of the members participating in the market, as opposed to the more traditional static path calculation.

6.4.1 Responsibility

6.4.1.1 Resilience

To initially judge that the BSP providers chooses the best quality services, the profitability of the BSP's over the initial run was analysed and the results shown in Figure 6.10. For this run the profit margin was set to 30% which was arbitrarily chosen as a overly high percentage, which should ensure that profitability was maintained even if there was a high percentage of failure. The graph shows that despite the verification of both service recovery and the move towards steady states of the individual SSP's the BSP's were still losing money. This means that despite the the arbitrarily high percentage, the BSPs were either experiencing a great number of failures, requiring a large number of re-provisions or that the cost of the individual re-provisions was very high.

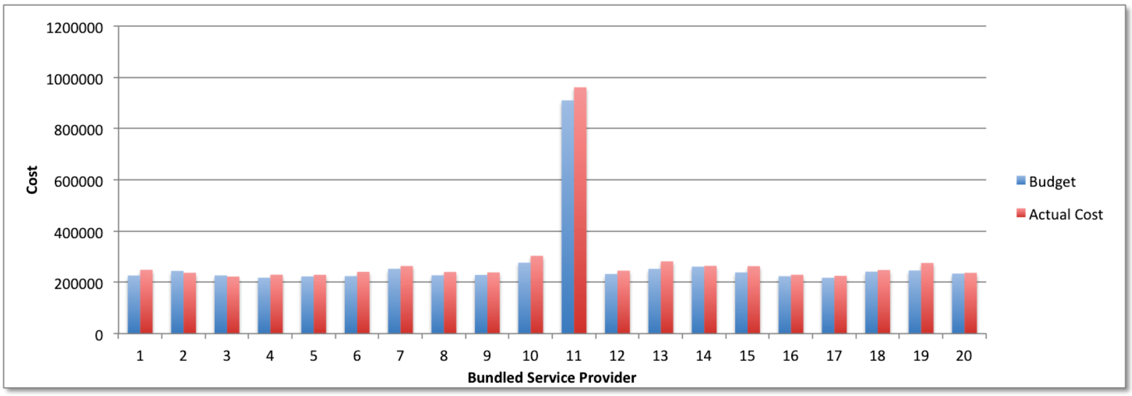


Figure 6.10: Initial Budget vs Cost for each BSP.

Further analysis showed that it was the latter, although the vast majority of contracts succeeded the re-provisioning of some of the services became very costly. Figure 6.11 shows a 1 in 10 sampling of one providers contracts profitability with a budget of 30%. While the majority of services maintain the 30% profit, when the service requires re-provisioning the costs can become astronomical. Further analysis and reflection was performed at this juncture with the results of this analysis presented below.

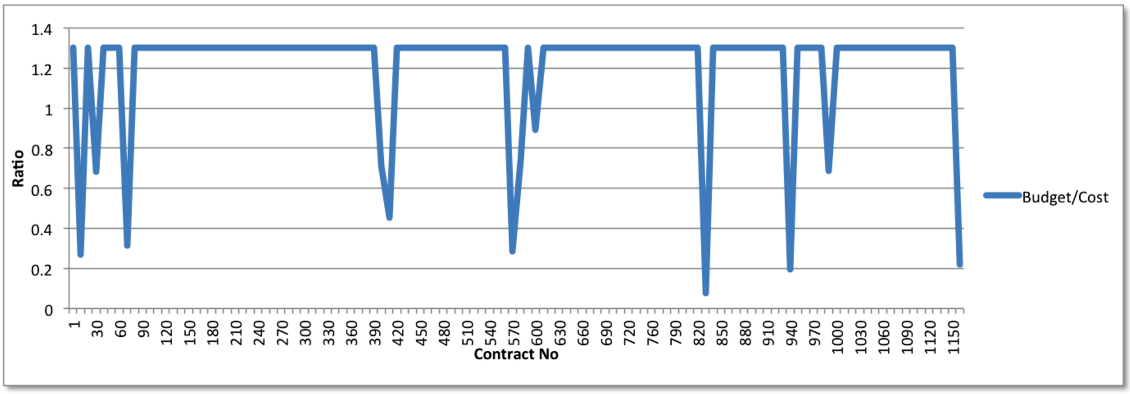


Figure 6.11: Budget vs Cost ratio for one BSP at a sampling ratio of 1 in 10.

6.4.1.2 Path Re-provisioning discussion

As the cost of re-provisioning in the case of failure has the potential to be a significant cost to the service, the mechanics of re-provisioning, and initial assumptions were examined. Firstly the mechanics, in the simulation, if a particular edge or provider fails the reserve functionality of re-provisioning is handled in two stages. The first stage is to look for SSP's that exist at the same node as the failing SSP (typically SSP0 will 'failover' to SSP1).The BSA then sends a message to the second SSP at the node to request whether or not the second SSP has the required capacity for this service, if the SSP responds affirmatively the BSA single service module re-provisions with that SSP. If, however, that SSP has not the required capacity then the BSA's single service module runs Dijkstra's optimal path algorithm to find an alternate path. The weightings for the Dijkstra's algorithm is determined via a combination of the BBN's risk assessment, and the cost of provisioning with the new providers. If all of the providers that are in the new path respond that they have enough capacity, then the BSA negotiates with the SSP's and reserves the new path. If any of the providers are not free then the Dijkstra's algorithm is run on a sub-graph that does not include the not free providers. This structure can be seen in the sequence diagram Figure 6.12 below.

This method of reservation is in line with the worst case initial assumptions that were made about the interaction between BSA's and SSA. The assumptions that were questioned were:

1. That the SSP would only release information that is relevant to the service they are provisioning and would not release information with regards to its whole capacity.
2. Accepting total responsibility of the service requires that the BSA to continue re-provisioning indefinitely under all circumstances.

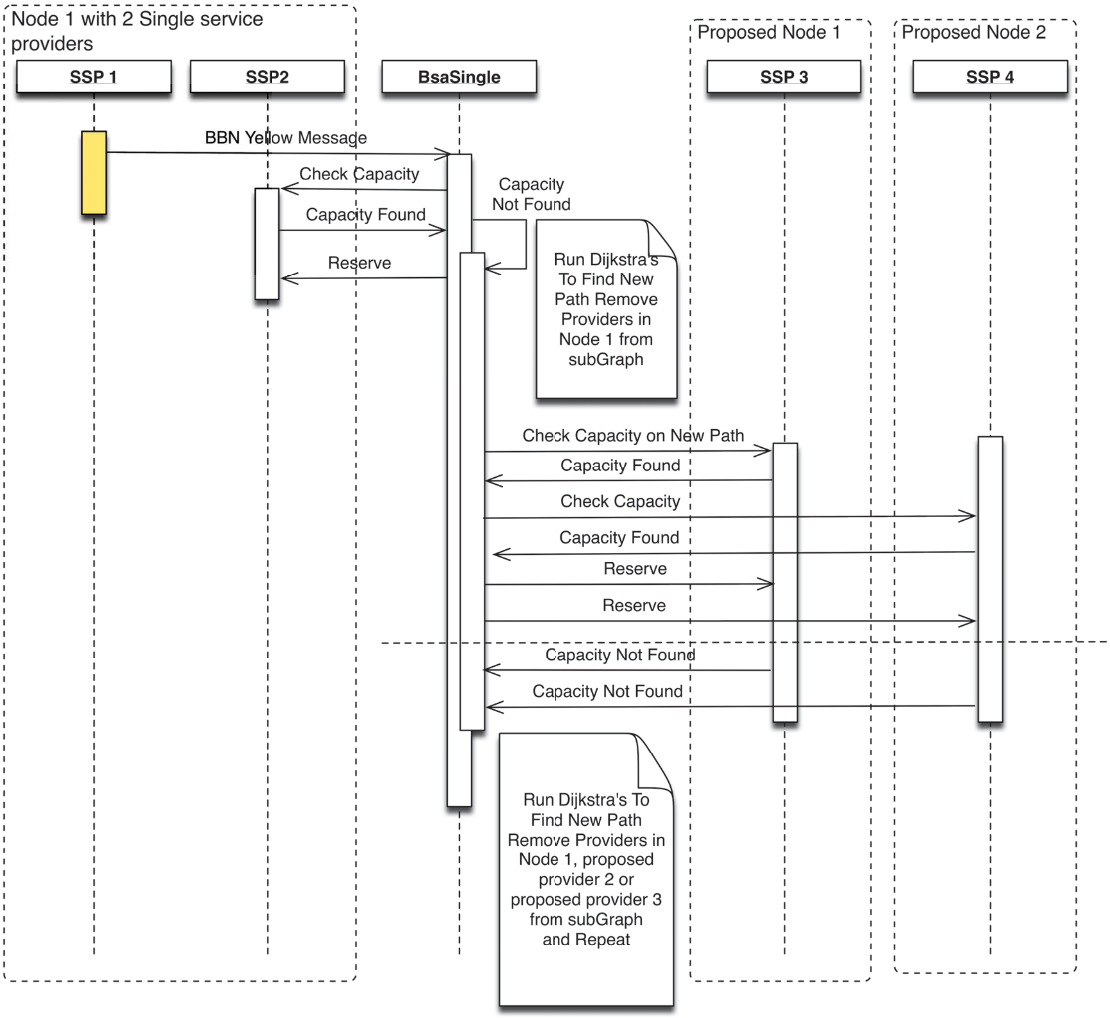


Figure 6.12: New reserve sequence.

3. That the BSA will absorb the cost for the service if it fails.

For assumption one, the author interpreted the assumption by limiting the information by the SSP to the cost of the service and a yes/no response to whether or not the SSP deemed that it has enough space for the contract.

However, the analysis showed that this level of information was insufficient to the needs of the BSA in the simulation. Utilising this simple yes/no response resulted in cascading failures, as multiple BSA's renegotiating for new paths, from one provider failure, would likely choose the same optimum providers for similar contracts, resulting in the overloading of the newly chosen providers.

To address this issue, this worst case assumption on the level of information between BSAs and SSPs needs to be relaxed. Upon reflection it was decided to allow some general information regarding the utilisation of the whole SSP to be communicated to the BSAs upon request. In the simulation this translated into allowing the find path functionality of the BSA's to request the utilisation of the SSP's in the path.

With this alteration to the interaction of the BSAs and SSPs, the mechanics of re-provisioning still has the potential to result in 1) low profitability for the BSA, and 2) a risk of further failure. As previously discussed, the BSA will always choose a provider at the same node as the initially failed provider, if available, regardless of cost. This potentially leads to the undesirable situation where the cost of continuing the service is so expensive that the total profitability of the BSP is adversely affected. At the same time, the load on the new provider can become so high that the initial problem of cascading failures can still occur, resulting in even greater costs.

This analysis led to the reflection that the assumption that re-provisioning under all circumstances was flawed. As some of the SSPs relied on unreliable SSRs, continually re-provisioning to those providers came with an increased risk of failure, while only increasing the cost to the BSA. Upon analysis it was decided that there must be a point at which the BSA determines that the service is not viable and cancels the service. Further reflection determined that this is a situation that has a basis in current SLAs between service companies. There are penalties involved for any company level downtime that exceeds the contract specification. Indeed without penalties such as service cancellation, the use of a responsible BSA encourages collusion by unreliable SSPs, as the SSPs do not face any penalties for failed services, where as the BSP does not receive any financial recompense.

In the simulation this was implemented via a maximum allowed re-provisioning cost, which was calculated at half the initially predicted cost of the service. Following this reflection period the simulation was run with a focus on the validation of the improvement of reliability and profitability with the use of risk management. Concurrently to this next series of simulations the simulation failures were analysed to determine why they failed and if they did so for reasons other than cost.

6.4.1.2.1 Failure It was discovered that there was a number of failures in the simulation. Initially, there were only one possible failure type, which was due to the BSA being unable to determine an available reserve path on re-provisioning. Providers in the center of the connection may re-provision to the other provider in the node and the possibility of new paths, as shown in Figure 6.12. However, for the endpoints of the contract there is only the possibility to re-provision to another provider at the same node. Therefore, in the case of failure, the endpoints are considered separate to no path failures.

Cause	No Contracts
No Path	4
Endpoints	59
Fail Cost	86
Unknown	0

Table 6.2: The causes of Failure in the Simulation.

With the addition of the Failure due to cost scenario, the results of the simulation were analysed to determine if any failure occurred that was unaccounted for by these three types of failures. The results of this analysis is shown in Table 6.2.

The Table shows that there were no failures that were not accounted for by the three failure types. More importantly the Table shows that the vast majority of failures was caused by either the endpoint scenario, or the cost scenario, indicating that it was very rare for the BSA to not be able to determine an alternative path for re-provisioning when needed, despite the limited resources.

6.4.2 Reliability and Risk Management Cycle 2

The simulation was run twice, once with the use of the SIMS for risk management in both initial contract negotiation and re-provisioning and once with no risk management. For no risk management the BSAs instead chose paths based on minimum price. It was expected that due to the implementation of the risk management system that the profitability of the BSAs using the SIMS would be greater than the BSAs utilising the minimum price strategy.

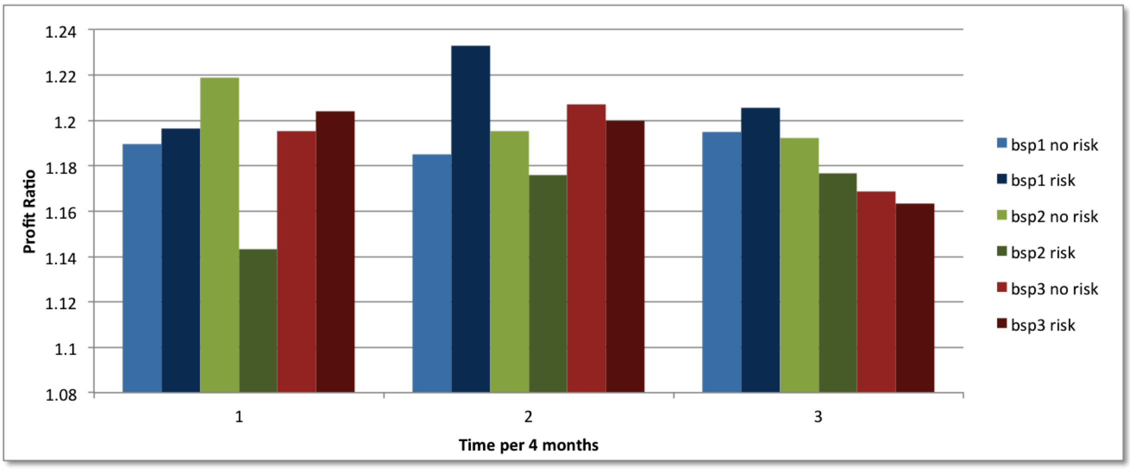


Figure 6.13: Profitability of BSAs under risk management and under minumum cost strategy.

The graph in Figure 6.13 shows the results of three providers profitability with risk management and without. The graph shows that although BSP1 with risk management demonstrates greater profitability than its no risk counterpart, BSP2 does not in any of the 4 month periods, while BSP3 does so only in the beginning. For the rest of the BSP's the spread was similar, indicating that the risk management had no appreciable affect on the reliability of the services and the profitability of the BSP.

6.4.2.1 Risk Management Analysis and Reflections

An initial examination of the cause of failures in the risk management run revealed that there were several issues with the initial simulation set-up with regards to the application of the SIMS.

1. Firstly, In the input to the historical service performance node of the BBN.
2. Secondly, in the risk judgement requirements per quality path.
3. Thirdly, in the conditional probability of the BBN.

6.4.2.1.1 BBN Input : Analysis of the BBN input showed that although services experienced problems and failed as indicated by the lower profit of the BSA (Figure 6.13) the far greater number of green messages, resulted in overly optimistic judgements by the BBN. Figure 6.14 shows the historical service performance input for one BSP in the market after the BSP had experienced several failures. It can be seen that although the BSP experience failure of services, the resulting risk judgement percentages remain very high.

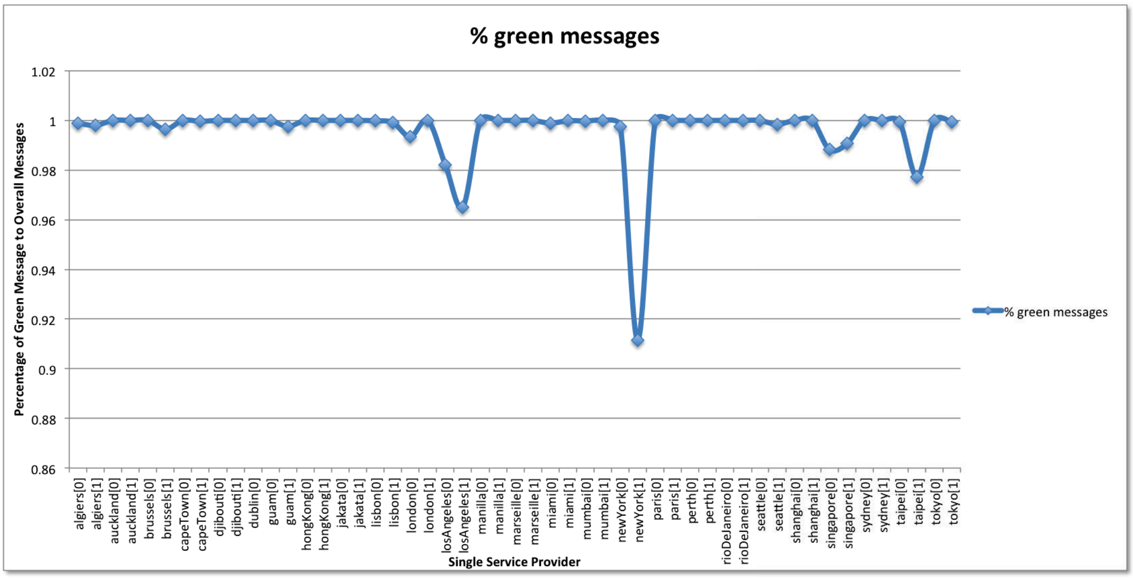


Figure 6.14: Percentage of green to overall messages as input to BBN.

Through reflection on the analysis, it was decided to weight the problem and failure messages to more appropriately reflect the severity of failure to risk management. The action of weighting the BBN input involved a series of mini-cycles to determine the optimum weighting for yellow, orange, and red status messages which form the input into the BBN. While the weighting of failure had to be severe, it was important that the weighting of orange messages did not overwhelm the input of yellow messages.

6.4.2.1.2 Risk Judgement per Quality Path The initial assumption for risk management in CARMA was that the risk would be judged per provider. The reason behind this initial assumption was that the reliability of service providers would be universally applicable across all quality levels. This assumption was based on the concept that with modern QoS management, networks experiencing difficulties will drop lower quality services first. However, in the simulation, while the failures due to load would affect the lower and medium quality services, they were less likely to affect the higher quality services. Concurrently, with the implementation of weighted inputs to the BBN, it was observed that the altered BBN inputs unfairly biased the BSAs against SSPs that failed in low quality contracts, when contracting later higher quality services. The observed effect in the simulation was that in markets of limited resources, cascading failures occurred throughout the initially chosen providers, which were then unfairly biased, which in turn led to failures of other providers. This is shown in Figure 6.15

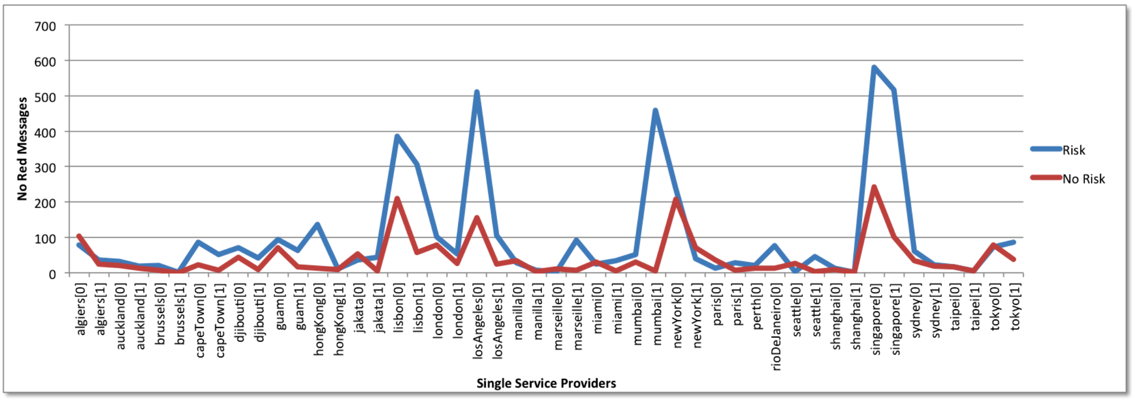


Figure 6.15: Failure in risk judgement vs no risk judgement (Based on the number of Red messages received).

To implement risk management at the individual quality path level involved two tasks. Task one was to implement the collection of status messages per contract quality and rebuild the BBN with the historical performance at the quality level when required. Task two involved altering the determination of the optimum path to better reflect the requirements of the contract quality level. The reasoning behind this change was the realisation that the BSP would have different priorities with regards to cost and risk at the different levels of contract quality. For High quality contracts, the BSP would prefer more reliable SSPs, where as for Low quality contracts the BSP would be more interested in cost. To implement this in the simulation, the inputs to the formula that was used to weight the edges of the graph was altered to incorporate the different quality levels. Initially all quality levels included this simple formula cost risk formula:

$$Cost * \lg((1 - Risk) * ScalingFactor) \tag{6.1}$$

As was explained in Section 5.4.8 the logarithmic risk was calculated as the risk percentage difference could be quite small. In the new algorithm however, the input to risk was scaled to different degrees, to reflect the difference in importance of risk to the weight calculation. This scaling also reflected the realisation that the difference in reliability for High quality services was much smaller than Low quality services. The scaling factors that were adopted are presented in Table 6.3.

Quality	Decimal	Scaling Factor
High	0.000001	10000000
Medium	0.00001	1000000
Low	0.0001	100000

Table 6.3: Scaling Factors for edge weights in optimal path algorithm.

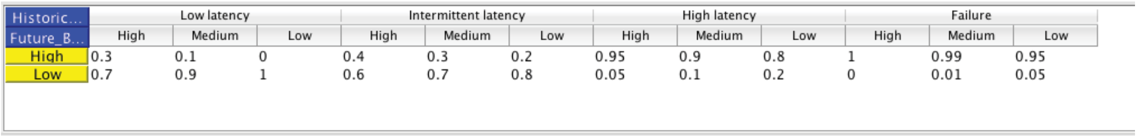
6.4.2.1.3 Direct BBN changes The third action came from further analysis of the outputs of the BBN with regards to the newly added weighting to the inputs. Originally the results were analysed as a basic sanity check as the creation of the BBN was done by the author (as the domain expert) before the model was completed. As can be seen in Table 6.4 the results showed that for the occasional judgement, the BBN output would rate the probability of choosing a contract with an unreliable provider higher than that of a more reliable provider.

Edge	Low	Intermittent	High	Fail	Output
hongKong[0] to singapore[0]	0.5694	0.0197	0.1027	0.3080	0.4708
hongKong[0] to singapore[1]	0.9269	0.0730	2.7446e-05	0	0.4637

Table 6.4: Erroneous BBN results for Edge.

This realisation led to an investigation of the BBN, and the fault was traced to the inference settings of the Customer Expectation node. This node was resulting in poor results in the Risk of Multiple Contract Failures node, which in turn was affecting the BSA Market Action output node. Upon reflection it was decided that this node was no longer needed as the assumption behind it, that the customer would wish consistency for High and Medium quality services, but would tolerate some small intermittent problems in Low quality services was appropriately handled by the division of the Historical Service Performance inputs into quality based path judgements.

At the same time it was decided to adjust the particular probabilities in the Risk of Multiple Contract Failures node, as the initial probability spread was judged to be slightly aggressive when concerned with providers that had experienced failure and yet had a low expectation of Future Bundled Requests. Figure 6.16 shows the new probability weightings for the Risk of Multiple Contract Failures, and Figure 6.17 shows the new BBN structure.



Historic...	Low latency			Intermittent latency			High latency			Failure		
Future_B...	High	Medium	Low	High	Medium	Low	High	Medium	Low	High	Medium	Low
High	0.3	0.1	0	0.4	0.3	0.2	0.95	0.9	0.8	1	0.99	0.95
Low	0.7	0.9	1	0.6	0.7	0.8	0.05	0.1	0.2	0	0.01	0.05

Figure 6.16: New probabilities for the Risk Of Multiple Contract Failures node.

Table 6.5 shows the new output and inputs of the BBN for the same contract, with the changes made for the risk judgement per quality path, the weighting of the inputs and the

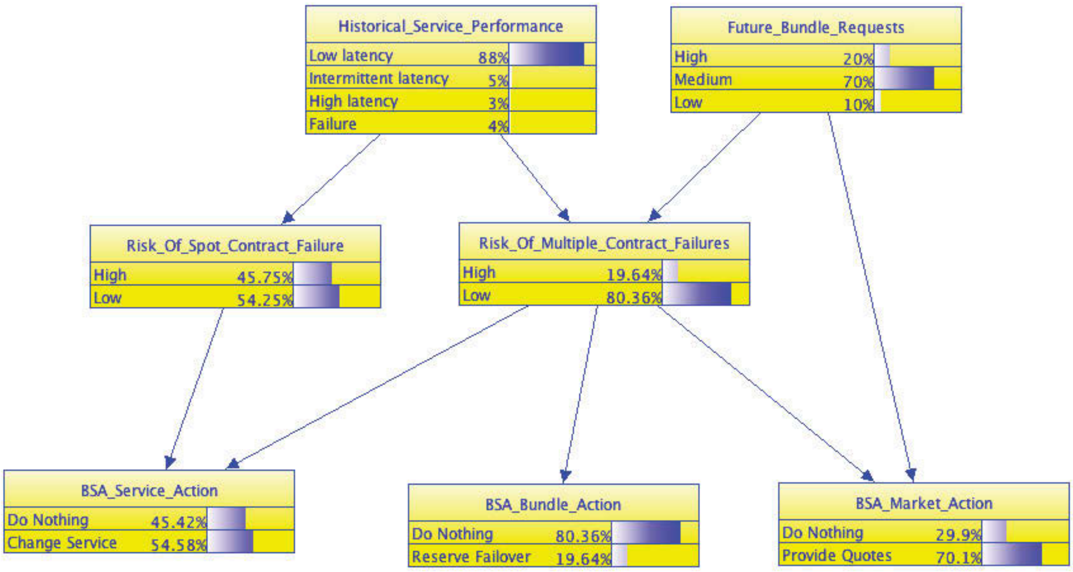


Figure 6.17: New Bayesian Belief Network for contracting new services.

BBN. The changes have resulted in not only the correction of the BBN output but also in that of the inputs.

Edge	Low	Intermittant	High	Fail	Output
hongKong[0] to singapore[0]	0.9945	0.0053	3.7770e-05	0	0.4809
hongKong[0] to singapore[1]	0.9909	0.0010	0.0079	0	0.4827

Table 6.5: Corrected BBN results for Edge.

Concurrently, it was also decided to utilise the Risk of Multiple Contract Failures node as the input to the edge weighting function, as opposed to the previous BSA Market Action node. The reasoning behind this reflection is that while the BSA Market Action node is concerned with the BSAs decision to offer a quote(or not) for the contract based on its judgement of risk of all of the providers that would be involved. However, as the simulation performs the risk assessment as part of the input to the Dijkstra algorithm, per provider, it was felt that the Market Action node, with its further conditional requirement against the future bundled requests was inappropriate for the risk assessment.

6.4.2.2 Further General Reflections on Risk Management

During the analysis period of the simulation risk management cycle another general assumption was challenged. Initially in the random contracts generated, the specified quality requirement was evenly distributed between High, Medium and Low. However upon reflection the even distribution seemed a poor reflection on the expected requirements of a QoS based dynamic marketplace. In any market based environment it is unlikely that users would choose an equal number of High quality services, with their attendant expense, and Low quality services. Rather, High quality services would be reserved for critical uses and the vast majority of service contracts would utilise a low quality contract.

During the analysis of this cycle it was also discovered that there was a programming error in the reservation and scheduling system of the SSPs. This programming error would lead to over-provisioning of SSRs for a certain small percentage of contracts, resulting in poorer performance for the services that were contracted. While the error was resolved, it highlighted the importance of the scheduling and reservation systems in the SSPs for the viability of the CARMA system.

6.4.3 Risk Management Cycle 3

To judge the effectiveness of the refinements to the simulation that was the result of the analysis and reflections of Cycle 2, the simulation was run with only one BSP to reduce the run time. Additionally, when using multiple BSPs, initially default settings for the BBNs are employed. Further the User Agent strategy in the simulation is to choose the BSP with cheapest quote. The use of multiple BSPs means that cheaper contracts involving less reliable providers are chosen because not all the BSP's SIMS have the necessary historical information. Limiting the number of BSPs to one creates a more complete performance map for the BSPs BBN. The simulation was then run under the new conditions and the results analysed to determine:

1. The number of actual failures was reduced;
2. The number of required re-provisions and failures was reduced utilising the risk management; and
3. and that the profitability of the BBN was increased.

The simulation was run under three separate conditions, with no risk management, with an untrained BBN and then with a trained BBN. A trained BBN refers to the concept that the BBNs historical performance node being preloaded with the resultant values of the previous untrained BBNs simulation run. In all three runs, the request contracts generated were identical with regards to endpoints and duration.

6.4.3.1 Reduced Actual Failures

Firstly the three simulation runs were analysed to determine that the number of actual failures on the SSP's was reduced under risk management conditions, compared to a no risk strategy.

Table 6.6 shows the results of this analysis. While in the case of both risk managed and not risk managed runs, the High quality serviced contracts experienced no failures, there was a significant difference in the number of failures in both Medium and Low contracts. Further, while the trained BBN experienced a couple more failures in the Low quality services, it prevented any in Medium.

Run	High	Medium	Low
No BBN	0	78	1288
BBN Untrained	0	1	226
BBN Trained	0	0	231

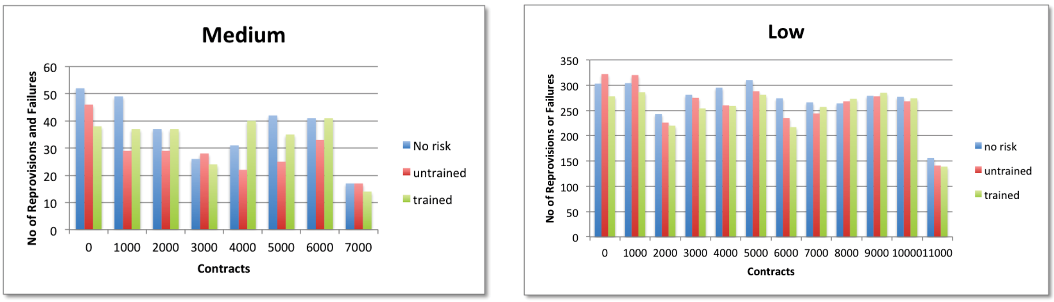
Table 6.6: The number of Total Failures in Each Simulation Run.

6.4.3.1.1 State Machine based performance judgement discussion Further analysis and reflection on the reasoning behind the vast difference in actual failures revealed that, as expected, the algorithmic performance judgement in the SIMS was responsible for re-provisioning and failing over to secondary providers before the primary provider had failed.

Observing the load and queue sizes at the time of fail-over showed that in some cases fail-over occurred much too soon, however, in other cases it occurred in a much more timely manner. Reflecting on the results and analysis determined that there is a need for greater refinement in the current performance judgement of the SIMS. However, that is left to future work.

6.4.3.2 Reduced Number of Re-provisions and Failures, and Increased Profitability

The second goal in this risk management cycle was to judge the effectiveness of the BBN in choosing providers that reduced the risk of requiring re-provisioning. To judge this, the results of three runs (no BBN, untrained BBN, and trained BBN) were analysed to see the number of re-provisions required, as well as failures (from the BSA's point of view).



(a) (b)

Figure 6.18: No of Contracts Failed or Re-provisioned per 1000 contracts.

Figure 6.18(a) and Figure 6.18(b) Shows the number of re-provisions or failures per thousand contracts in both Medium and Low. As there were so few in High, it is not represented in a graph, rather it is presented here as Table 6.7.

The Figures 6.18 and Table 6.7 show that though overall the number of re-provisions and failures was reduced under risk management conditions, the degree of difference was for the most part very slight. Indeed under the conditions of a trained BBN where one

Run	No of Failures
No Risk	4
Untrained BBN	0
Trained BBN	3

Table 6.7: No of Failures in High.

would expect to see greater reduction of re-provisions the results showed that for Medium contracts, as the simulation continued, the total number of re-provisions and failures were almost on par with no risk management at all.

This poor risk management was reflected in the profitability of the BSP. The graphs in Figure 6.19 show that for Low contracts the risk management resulted in greater profitability for the BSP. However for High contracts, the provider choices which were the result of the risk judgement of the trained BBN resulted in a small number of re-provisions, reducing the profitability. More significantly for Medium contracts the trained BBN was resulting in reduced profitability for the BSP as the simulation continued. This indicated that, over time, the BBN was making worse decisions regarding provider choices. The result of which was both poorer reliability for the user and a profit margin on par with no risk management for Medium contracts.

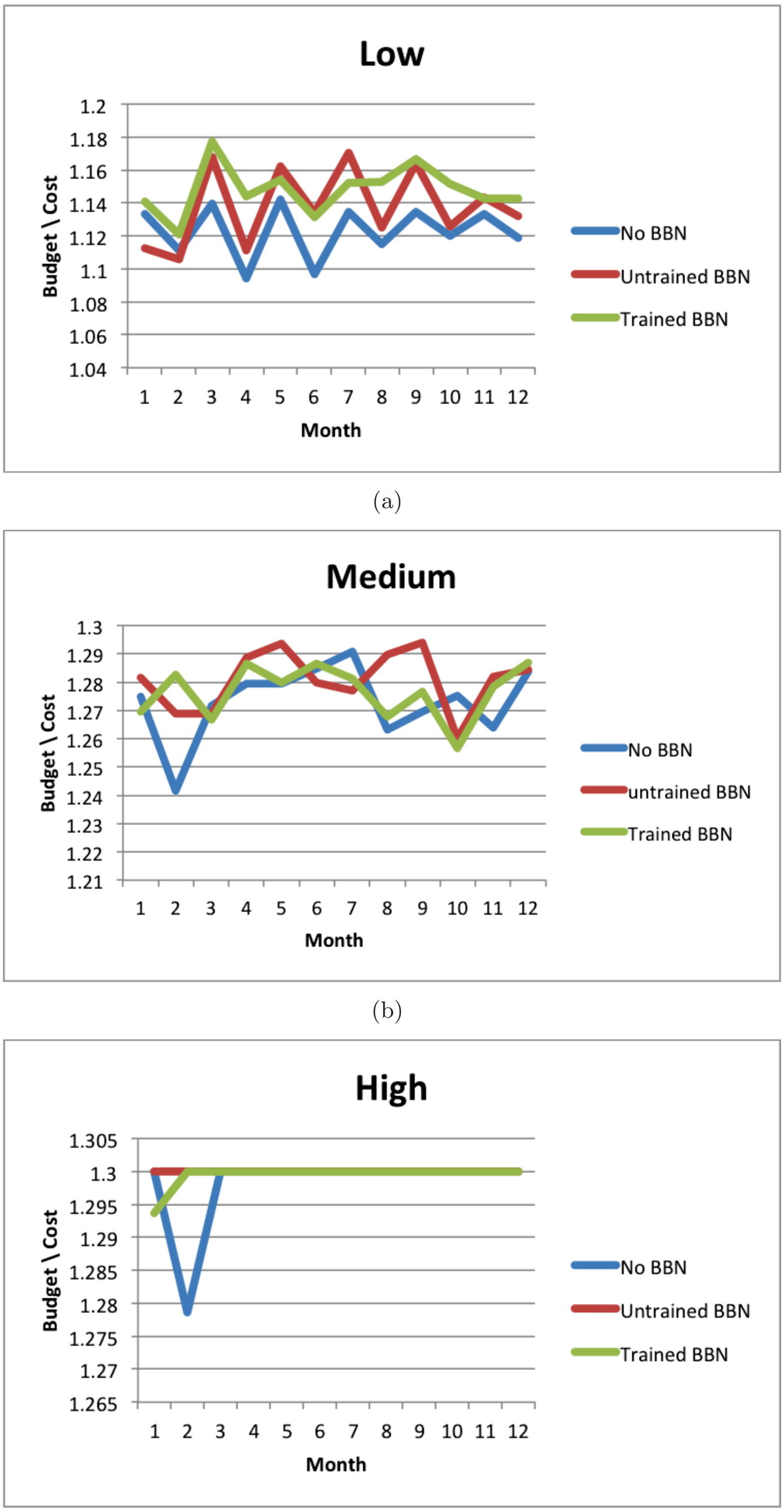
6.4.3.2.1 Risk analysis Analysis of the historical performance node inputs for the BBN, revealed that the cause of the poor provider judgements was the result of four factors.

Firstly, while the output of the BBN was correctly judging the risk of contracting with each individual provider (based on its inputs), the formula that was used to weight that judgement with regards to the risk (Section 5.4.8) was still being incorrectly applied. The changes introduced to the BBN with regards to the use of the Multiple Risk of Failures node as the output of the initial formula was devised to contend with the small risk percentage difference of the original output. However, the changes to the BBN resulted in much larger differences under heavy load conditions. Further analysis showed that these differences were set in three bands, 60% - 70% for high load conditions, 80% - 90% for medium load conditions and 90%-100% for no load. This in turn caused less differentiation between outputs in congestion scenarios.

Secondly, the implementation of congestion pricing, initially set at a flat rate of increase, was reducing the percentage difference of the price. This was resulting in a less reliable edge weighting, under load scenarios, to Dijkstra's algorithm.

Thirdly, the initial random spread of price differences between unreliable and reliable providers was resulting in overly large differences for some providers, and really small differences for others.

Finally for High quality contracts the BBN was giving a judgement of lesser risk to unreliable providers than reliable. The reasoning behind this judgement involved a combination



(a)

(b)

(c)

Figure 6.19: Profit Per month at each Quality Level.

of factors. One, as unreliable providers are cheaper, the weighting formula, which is outside of the BBN and combines the risk judgement and cost, will initially choose that provider in an untrained BBN that has the same initial judgement of risk. Two, the priority of failure in the SSR's, means that low quality services are made to fail first, under heavy load, then medium and lastly high (shown in Table 6.1). This in turn results in High contracts that never fail under circumstances where there are a mix of quality level services on the SSR. Three, the service performance messages are kept, and form the basis of the historical performance BBN input. As the High quality services only rarely fail on unreliable providers, performance messages indicating that the provider is reliable would continue to accumulate for unreliable providers, which over time results in a judgement of lesser risk. This judgement of lesser risk results in more contracts, continuing the cycle.

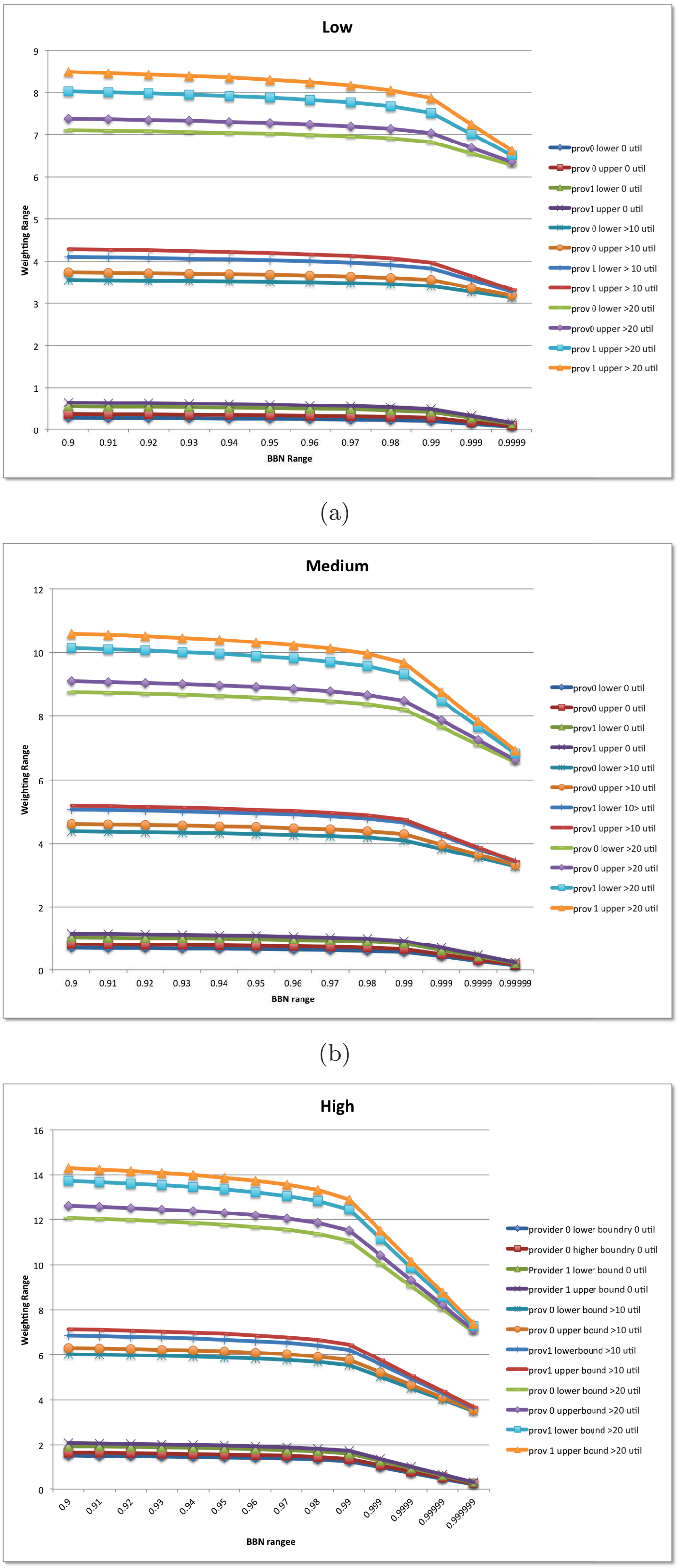
6.4.3.2.2 Risk Reflection and Planning Reflecting on this analysis led to some further fine tuning of the BBN inputs, the inputs to Dijkstra's algorithm and a change in the congestion pricing implementation.

Firstly, the inputs to the weighting formula were altered to remove the inconsistencies in the edge weightings caused by the different bands outputted by the BBN. Secondly the congestion pricing was changed to a percentage increase, and thirdly the price difference between providers was set to a fixed difference.

The results of these changes to the pricing structure and weighting formula can be seen in the non linear graphs of Figure 6.20. The graphs show that for each quality level the weighting gap between the providers is now more uniform and that the risk judgement has a greater effect on the weightings, subject to the providers congestion.

Lastly, the issue of High quality contracts getting unreliable risk assessments due to the greater overall reliability of High quality providers, required some changes to the initial BBN settings. While the previous changes are refinements to the system, this change to the initial settings could be considered changing the simulation to suit the designers needs. However, it was reasoned that this should be allowable as in all cases the initial conditions of the BBN, indeed the design of the BBN would be the initial responsibility of a domain expert. As such he or she would draw on previous knowledge about the Single Service Providers in the market to design the initial BBN. Indeed it is believed that such refinements would be continually taking place in order to obtain competitive advantages against the other market participants.

It was decided therefore, on the initial set up of the BBN to give a slightly better judgement of historical service performance to the more reliable providers (provider 1). The new inputs are shown in Table 6.8.



(a)

(b)

(c)

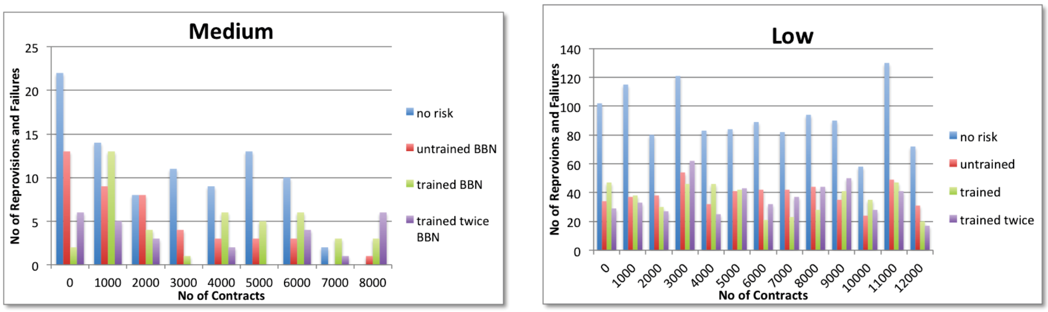
Figure 6.20: Range of Edge Weightings for Reliable and Unreliable Providers in all Quality Levels.

Provider	Low	Intermittant	High	Fail
0	0.96	0.03	0.01	0
1	0.985	0.01	0.005	0

Table 6.8: New initial inputs to Historical Service Performance Node of the BBN.

6.4.4 Risk Management Cycle 4

The simulation was then run another three times under similar conditions to the last cycle, namely under no explicit risk management, an untrained BBN and a trained BBN. Additionally the simulation was run one further time with the results of the trained BBN, being the initial conditions. The results were then analysed with regards to total re-provisions and failures, and profitability.



(a) (b)

Figure 6.21: No of Contracts Failed or Re-provisioned per 1000 contracts.

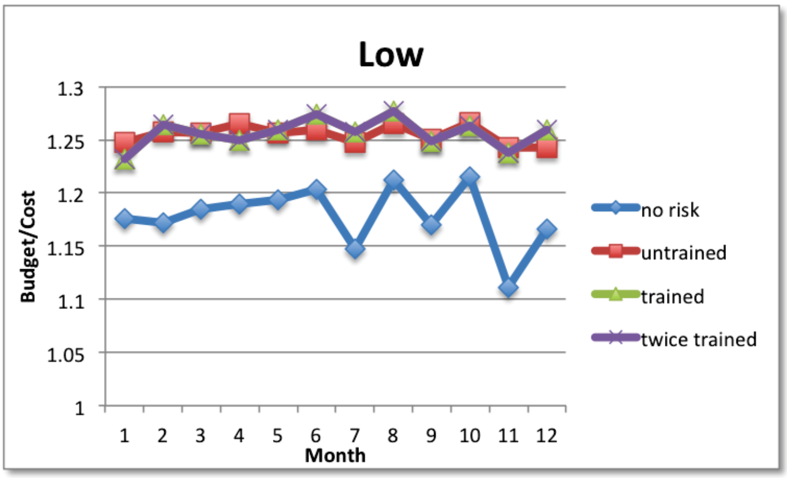
The Graphs in Figure 6.21 shows that the refinements made in the last cycle, reduced the number needed re-provisions and failures considerably in Low quality services (Figure 6.21(a)). In the Medium quality services the total number of re-provisions was also reduced significantly over the previous run.

Run	No of Re-provisions
No Risk	0
Untrained BBN	0
Trained BBN	13
Trained BBN 2	0

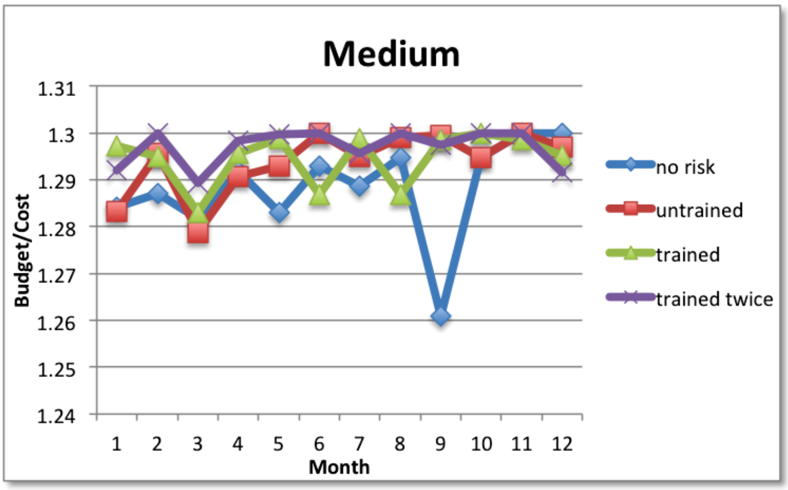
Table 6.9: No of Re-provisions in High.

However, in the High quality services, the trained BBN increased the number of needed re-provisions shown in Table 6.9. Though it is important to recognise that it experienced no actual service failure.

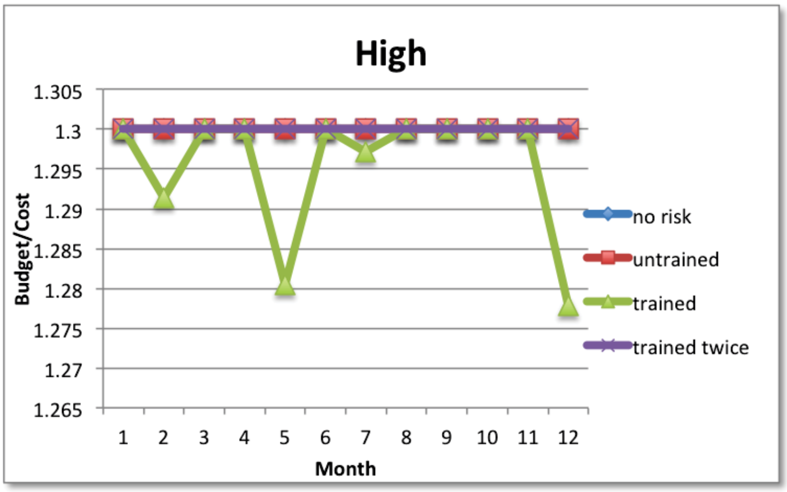
Looking at the resultant profitability graphs in Figure 6.22 shows that the reduction of re-provisions and failures resulted in an expected greater improvement to the profitability of the BSP (Figure 6.22(a)). Interestingly it also shows that the use of the trained BBN and the further trained BBN does not significantly improve the profitability, suggesting



(a)



(b)



(c)

Figure 6.22: Profit Per month at each Quality Level.

that the priorities of failure on the SSR and the limited number of resources to re-provision with, formed an upper bound on the profitability for low quality services.

However, the Medium contracts in Figure 6.22(b) did show an increase in profitability with a trained BBN. Additionally in the second trained run the overall profitability continued to improve. However the results showed that under the new conditions, the trained BBN experienced problems requiring re-provisioning in High quality contracts.

6.4.4.0.3 Analysis and Reflection Further analysis showed that in all cases the problems in the High quality services ultimately occurred due to two factors. Firstly, the problem highlighted before in Section 6.4.3.2.1 where the overall reliability of High services resulted in lower risk judgements for unreliable providers, was compounded by the initial refinement shown in Table 6.8. The refinement meant that the the low quality contracts, which constitutes the greater majority of total contracts, more frequently chose the more reliable providers. Additionally, the expected failures in unreliable providers reinforced the judgement. With the low quality contracts either reserving completely or causing greater congestion pricing on reliable providers in the simulation, the path options for the High quality contracts, was consequently limited.

When the simulation was run again however, and the BBN had definite judgements regarding the reliability of the unreliable providers, the situation that resulted in this re-provisioning was completely avoided.

6.4.4.1 Refinement of Risk Management

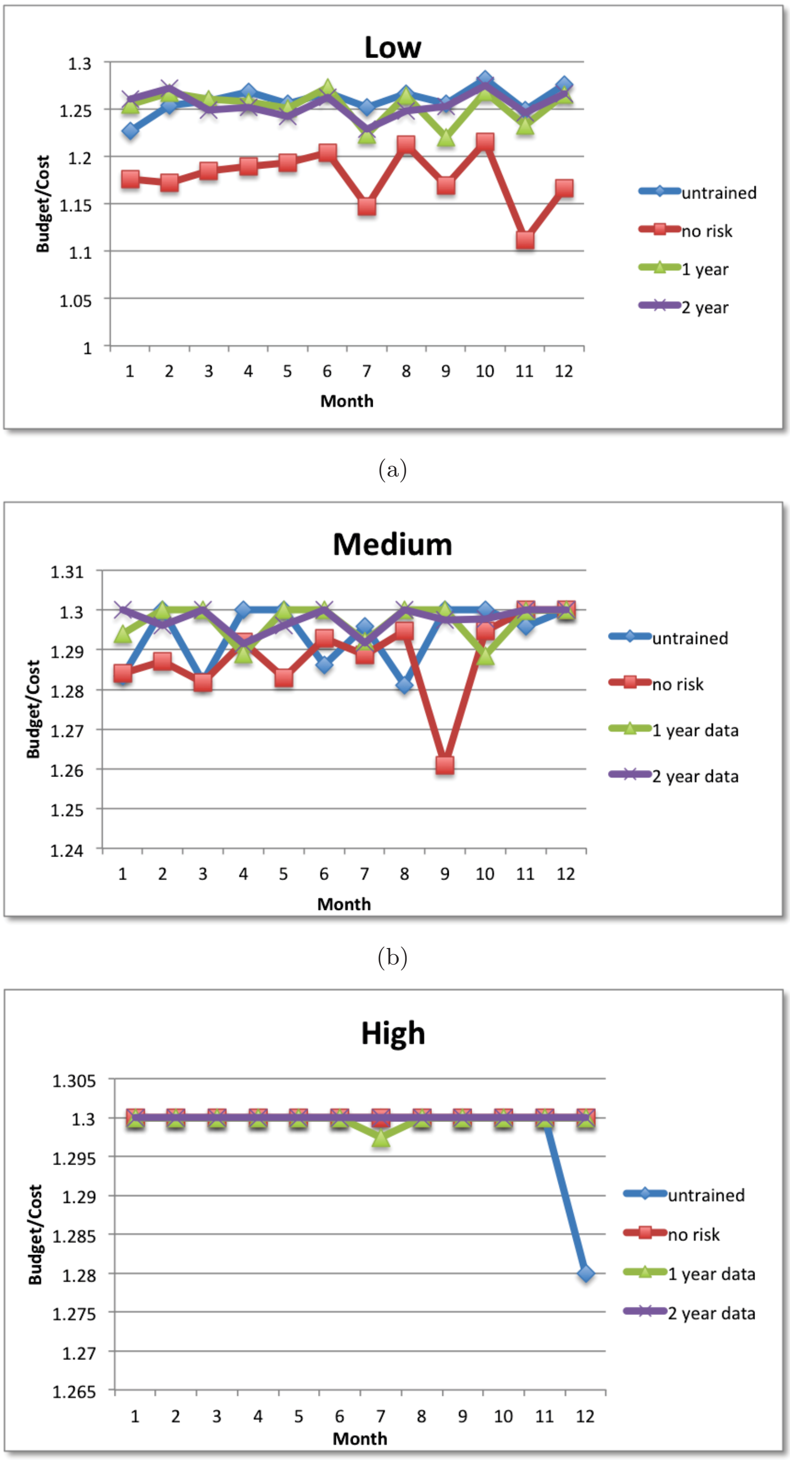
Some further refinement was attempted to address the issue of the overall reliability of High quality services affecting the risk judgement. This involved a greater bias towards more reliable providers in the initial historical service performance settings in the BBN. The new initial values were calculated by finding the greatest difference between received green messages for reliable and unreliable providers after two runs (untrained and trained) and applying this formula to achieve the new values of the Historical Service Performance node:

$$\text{intermittent latency} = \left(\frac{\text{provider 0 green messages}}{\text{provider 1 green messages}}\right) \times 2 \tag{6.2}$$

$$\text{high latency} = \frac{\text{intermittent latency}}{2} \tag{6.3}$$

The results of this formula was applied to the untrained BBN and the simulation was run a further three times, with the final BBN state of every run becoming the input to the next run. The graphs is Figure 6.23 show the profitability results for these runs.

The results show that there is improvement for High quality contracts, with only one needed re-provision in the first untrained run, and one re-provision in the first trained run (Figure 6.23(a)). Additionally there was slight improvements in the overall profitability



(a)

(b)

(c)

Figure 6.23: Profit per month at each Quality Level.

Risk cycle	Quality	untrained	trained	trained twice
Risk cycle 4	High	1.3	1.295730534	1.3
	Medium	1.293800224	1.294564703	1.29699834
	Low	1.25467557	1.256161296	1.259496844
Further Refinement	High	1.298467831	1.29978613	1.3
	Medium	1.293793276	1.296785852	1.297517582
	Low	1.258891154	1.252527575	1.2542914

Table 6.10: Overall profitability per quality contract.

in both runs in either the Medium and Low Contracts as the BBN continued learning, which is presented in greater detail in the Table 6.10.

Overall the further refinement to the initial conditions improved the profitability, and by implication the reliability, of the Medium contracts and the High contracts after the BBN had been retrained. While continued refinements could potentially remove the issue of slightly poor risk judgement in High quality services, the clear improvements in both reliability of service contracts and profitability demonstrated in the last two cycles clearly indicate the validity of risk management with regards to service provisioning and management.

6.4.5 Increased Utilisation Cycle 5

The final cycle focused on evaluating the effect of CARMA on utilisation. To understand the differences in utilisation the load on the SSP's resources was examined. In order to compare the utilisation of CARMA compared to traditional traffic management, the simulation was revised to be representative of current global communications. This included:

- Removing the Bundled Service Provider and its agents.
- Set the weightings for the path finding Dijkstra' algorithm to 1.
- Splitting the contracts evenly between the unreliable and reliable providers.

The contracts that were created were identical in both runs with regards to end points and service type. Additionally, the Single Service Providers maintained their reservation and scheduling systems. The result of this was that while most contracts with identical endpoints would always choose the shortest path (with the least number of providers), if the providers are fully reserved, the simulation would choose a new path, mimicking in abstract way the behaviour of modern routing. The simulations were then compared based on the average load on the SSP's with regards to simulation time (sampled every minute). Figure 6.24 shows the results of this simulation. The labels 'no risk' and 'new risk' represent the traditional usage and the latest iteration of risk management respectively.

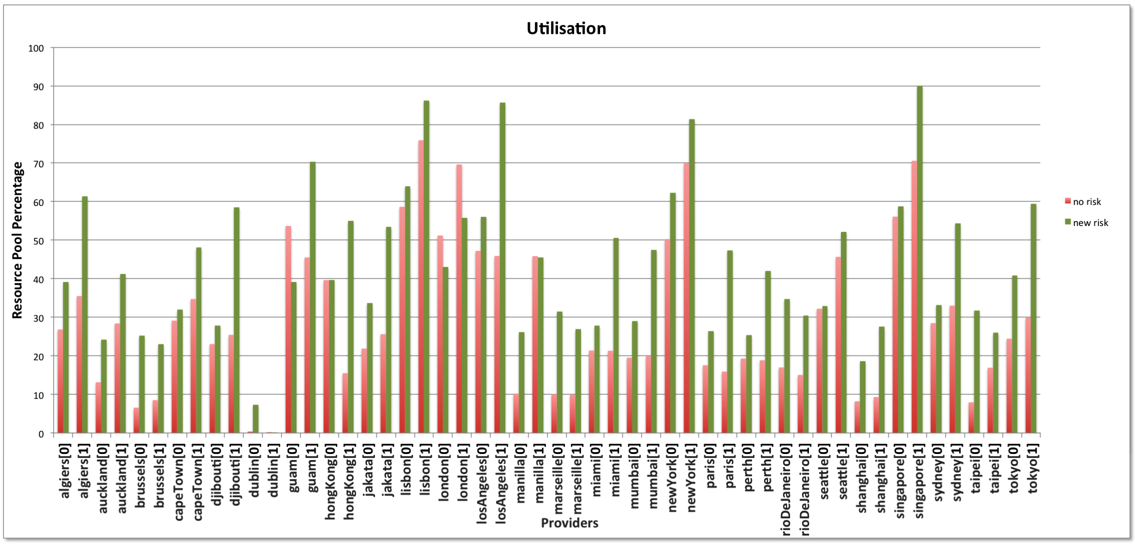


Figure 6.24: Utilisation of Traditional usage vs CARMA.

The graph in Figure 6.24, clearly demonstrates that under the conditions specified above the utilisation overall is significantly increased with the utilisation of CARMA. Additionally, the utilisation per provider is in almost all cases also much greater.

6.5 Discussion

This chapter has attempted to validate the Design of CARMA as presented in this thesis. During this verification and validation a number of issues were raised that require some further discussion. Initially the design for CARMA held some assumptions that proved untenable via the simulation.

Firstly, the cost of finding a new path is often prohibitive in the simulation, as shown in Section 6.4.1.2. This means that if a provider is not available at a SSP node then it is likely that the re-planned path would require at least two or more providers to fail over, with the cost of getting those providers most likely being much higher than their base price.

However, this would not generally be the case outside the simulation as telecommunication companies are much more interconnected than is reflected in the model with only 26 Single Service Provider nodes. This can be seen even with the simplified map of deep sea network links in Figure 6.1, as there are more nodes and more connections shown than are represented in the simulation. Consequently the cost of failure would reduce as utilisation of the CARMA model increased. Secondly, while the results showed clear improvement in reliability and profitability with the utilisation of risk management, the cycles showed that significant modifications were needed to tune the risk management component. These modifications in the simulation also have an impact on the design of the system.

Specifically with regards to the measurement of failure indicators, the method used in the simulation to weight the incoming messages implies that there is a need for some kind of preprocessing of the aggregated monitoring information received by the Bundled Service Provider and their agents. This preprocessing could be included in the Bayesian Belief Network or external to it as in the simulation.

Another point of discussion raised by the research cycles is that the development and fine tuning of the BBN is critical for the successful management of risk in this environment. Indeed it is foreseeable that the BBN or whatever method of risk management is utilised by individual BSPs could become part of the core business requirement for the Bundled Service Providers, constituting the competitive edge required in the marketplace.

Additionally the focus of this work has been on the failure of services due to load and congestion issues, and not related to the case of catastrophic failure due to unforeseeable power issues, total device failure, or human misconfiguration. While the design of CARMA included provisions to deal with such failure, as can be seen in the algorithmic performance judgement in Section 4.4.2, these failure types are completely unpredictable without intimate knowledge of the devices environment. This level of detail is currently unobtainable by any external entity. Therefore such failure cannot be predicted by the SIMS and not germane to the simulation.

The elements and design of the CARMA system in Chapter 4 was designed with scalability in mind, however the simulation environment limitations made the validation of scalability in the design problematic. Increasing the number of resources and agents increased the memory consumption of the simulation beyond the capabilities of the simulator. With the simulation design parameters of the resources limiting the resources to four concurrent contracts, the number of contracts that could be serviced was limited by the number of resources that could exist in the simulation before memory issues rendered the simulation unresponsive. This issue meant that for scalability validation the bottleneck was in the simulation environment rather than the design.

Overall, the results of the simulation indicate potential for CARMA in the management of dynamic end-to-end services, demonstrating improved reliability and resilience for contracted services, and increased utilisation and efficiency for the individual Single Service Providers.

Chapter 7

Conclusions and Future Work

7.1 Summary

The work presented in the preceding chapters of this thesis focused on the issues pertaining to the exploration and implementation of a system for performance management of bundled services in a multi-domain environment.

Chapter 2 presented a literature review that attempted to arrive at the reason why the continued improvements in the management of telecommunications and cloud environments have never completely addressed the basic requirement of end-to-end service management. This was achieved by first examining the fundamentals of network management and exploring the evolution of research focus as both the technology and utility of telecommunications changed.

In the beginning network management was focused on the network as a whole. As a result the research and technologies developed were focused on the management through a centralised platform (Section 2.1). As networks expanded in both size and complexity, there was a shift in focus towards the management of services in the network. This shift was the basis for a succession of proposed architectures aimed at proactive management. A general discussion into initial Service management aimed at managing quality of service across domains demonstrated that due to the immaturity of the technologies at the time, the proposals could not move forward (Section 2.3). Another concurrent approach was the proposal of Policy Based Management, with the goal of managing services and networks through the integration of business rules.

Additionally, to overcome the limitations of centralised management in telecommunications researchers focused on Autonomic management, with autonomous agents guided by policies, as a scalable decentralised approach to management. More recently this research has attempted to build a federation through the hierarchical cooperation of autonomic management systems for the purpose of end-to-end service management (Section 2.4).

Concurrently in the field of Cloud management researchers have also utilised the concept of Federation in the management of scalable cloud services. The concept of federation in cloud computing has focused on toolkits and agents for the deployment and oversight of cloud services in a wide variety of public and hybrid clouds (Section 2.5.1). Cloud research has also attempted to address the need for a dynamic marketplace for service interaction, through the use of strategies such as auctions and bargaining (Section 2.6).

However, both Federation through cooperation and Federation in cloud services fail to completely address some fundamental issues with regard to end-to-end service management. Cooperation at the level of autonomic systems for example is a complex endeavour, requiring an overwhelming amount of semantic management to integrate the multiple goals of the disparate autonomic systems. Where as the Federation of Cloud environments focuses too narrowly on the individual service management aspects of cloud services. Additionally there has been very little recognition of the concept of risk in the management of end-to-end services.

In response to the issues presented in the literature, such as the complexity and relative narrow focus of federation, and the limitations of current market proposals, an alternative framework for the end-to-end management of complex services based on competition was proposed in Chapter 3 (page 33) . The framework was developed by firstly identifying the fundamental requirements of the system, which was found to rely on the development of the concept of service *responsibility*. Responsibility in this context is defined as end-to-end management of individual service bundles above and beyond the management that would be provided by individual service providers. It was further defined as being accomplished through the mechanisms of *Risk Management* and *Resilience* which was defined as being beyond reliability, enabling services to recover from poor performance. Additional requirements of *Scalability* and *Efficiency* were also identified as fundamental to the framework.

In Chapter 3 the service independent requirements for contract specification were also defined (Section 3.3) and the marketplace mechanisms were also defined through an analysis of the system requirements (Section 3.4). The remainder of the chapter was primarily involved with identifying the entities and their relationships. As opposed to a federated approach the entities in CARMA, defined in Section 3.2, are competitive and have different goals and responsibilities. This is further defined through the architecture specification in Section 3.5.

With the defined architecture as a guide the design of CARMA was then addressed, specifically related to the functionality of the agents and their interaction across the market (Section 4.2). Additionally a method of contract specification and transformation through information models and Graph Theory was presented (Section 4.3).

From this design a model was developed to test the viability of CARMA in the market situation. The process of that development is described in Chapter 5 (page 78).

One of the key concepts to the CARMA design is the ability of the system to judge the risk of service contraction or management. In the literature presented there was very little acknowledgement of the role risk management could play in the management of bundled or complex services. In CARMA the requirements of risk management were initially defined through Section 3.5.2. A Bayesian Belief Network was developed in response to those requirements and implemented in the model (Sections 4.4.1 and 5.4.8).

Finally the model was verified and validated through a series of simulations based on a simulation study design described in Section 5.5. The simulations runs conformed to action research cycles and involved multiple refinements and developments that were the result of analysis and reflection on the outputs of the simulation runs. This is described in greater detail in Chapter 6 (page 96).

The final evaluations showed that CARMA is a valid approach to the management of complex end-to-end services. For the twin goals of Resilience and Risk Management the simulation showed that the system provides resilience in the management of poorly performing services. For risk management, the simulation showed that the use of a Bayesian Belief Network and performance judgement, decreased the likelihood of failure for any individual service and increased profitability for the Bundled Service Provider.

7.2 Research Questions Answered

In the introduction of this thesis a vision was presented that was to serve as the focus of the research. This vision was of:

> A marketplace in which all manner of complex services will be provisioned, and their performance managed across multiple domains, accepting the responsibility of each service. (page 5)

In the conduct of this research the author developed, refined and simulated a system of autonomous agents that negotiated in a marketplace and manage end-to-end services, providing resilience from the point of view of the user. The CARMA system was developed in a problem space whose boundaries were defined through an exploration of current management and research in the field of telecommunications and cloud computing.

The exploration of the problem space was guided by a number of research questions. Below are the research questions (taken from Section 1.2.2) accompanied by the references to the relevant sections of this thesis.

1. *In a multiple provider environment, what are the provider independent properties of a dynamic complex service?*
   The provider independent properties of a dynamic complex service were defined in Section 3.3, this thesis defined four general service properties, of *Path*, *Type*, *Time*, and *Quality*. While these properties are not intended to be an exhaustive list, they represent the most common properties of any dynamic telecommunication service. This was further expanded through the contract transformation in Section 4.3.

2. *What is the current state of service coverage in multiple provider complex service management?* The overview of the current state of service coverage in the multi-domain environment is the subject of the literature review of Chapter 2 (page 12). The conclusions of the chapter was that while both the telecommunication and cloud computing environments are attempting to address the requirements of a multi-provider environments, current approaches are limited by separate narrow focuses, and there is yet no proposal that attempts to provide coverage for the entirety of a modern complex services. Further, the proposals that exist rely on either too complex cooperation requirements, or market concepts that are deemed inefficient under the unique service conditions proposed by this thesis.

3. *What does it mean to be Responsible for a service? Specifically with regards to Resilience:*

   (a) *How can the performance, in terms of Quality of Service, of a service involving multiple independent providers be judged?*
   The author contends that for the performance judgement to be effective the agent responsible would need performance information received from the Single Service Providers management system as is discussed in Sections 3.5.1.3 and 4.2.4. The complexities involved in presenting a performance evaluation of the independent services in a bundle were discussed in Section 4.2.2 and the decision was reached that the aggregation of the monitoring messages would be presented to the Bundled Service Agent in four classifications, representing no issues, low issues, high issues, and failed, based on a simple colour classification that is utilised in industry.

   (b) *In the context of multiple service providers, working in concert to provide a dynamically created complex service, how can service resilience, be ensured?*
   At the level of interaction proposed by the CARMA system the Single Service Providers are deemed to be independent entities. Therefore CARMA proposes that renegotiation with alternative providers would be the primary method to ensure resilience for services. This renegotiation would work in concert with the reliability measures that would be undertaken by the Single Service Provider management system (Section 3.6) and ensured by the contract specification. The structure of this renegotiation is discussed in Section 4.2.2 and outlines four stages, that of Good, Reserve, Offload and Failover. These stages represent the concept that the Bundled Service Agent would be able to take proactive steps in the management of resilience, by identifying potential problems through its risk management system and actively migrating the service before total failure. The resilience structure was implemented in the system (Section 5.4.4) and its performance verified in Section 6.3.1. Further validation of the structure was provided in Section 6.4.1.1.

4. *With regards to Risk Management:*

   (a) *How can the risk of contracting a service be managed, when the service involves multiple individual providers?*
   The complexities of managing risk across a multi-provider environment was initially discussed in Section 3.5.2. The conclusion was a definition of integral requirements for risk calculation, the requirements included *performance information*, *user quality requirements*, and *Service functional quality requirements* This is further discussed in Section 3.5.3. This thesis then proposed that the Service Information Management System be implemented utilising a Dynamic Bayesian Belief Network, whose component input nodes represent the requirements listed. The initial structure of the BBN was presented in Section 4.4.1, and altered and implemented in the CARMA simulation in Section 6.4.2.1. The results of the simulation showed that the implementation of risk management had distinct advantages in terms of reliability and profitability and is discussed in greater detail in Section 6.4.4.

   (b) *How can the risk of maintaining the service under poor performance be judged?*
   The risk judgement of poor performance in a service already under contract was achieved using an algorithmic method that is discussed in greater detail in Section 4.4.2. The algorithm was implemented in the model and the results of its performance were judged in Section 6.4.3. The results showed that the devised algorithm was effective in judging the risk of service failure, preventing actual failure on the Single Service Providers. However it was conservative, in some cases leading to unnecessary reservations with their attendant costs.

5. *What kind of framework can provide resilience, risk management, and total coverage of complex services involving multiple independent providers?*
   The majority of this thesis was devoted to the development of CARMA, a system designed as a response to this question. By working alongside current management systems and taking place in a service marketplace, CARMA attempted to provide total responsible coverage to modern bundled services. To this end, Section 3.2 gave an overview of the entities and relationships in the system, and Section 3.5 provided the architecture. The design and functionality of the agents was discussed in Section 4.2 and a model, developed to the design was discussed in Section 5.4. To test the validity of CARMA, simulations were run based on a simulation design study discussed in Section 5.5. The validation was presented in Section 6.4. The results of this validation was that the implementation of CARMA demonstrated service resilience (Section 6.4.1.1) and risk management Section 6.4.4 and Coverage of the Service. Through risk management CARMA also displayed increased reliability of the services, and greater utilisation of individual service providers.

6. *What other elements are required by the framework to ensure effective responsible coverage of dynamic services in the multi-provider environment?*
   CARMA also explored an approach to contract specification and transformation over the different layers of abstraction. Based on the requirements specified in Section 3.3

the CARMA system proposes that contracts would be specified with the information model offered by the Tele-Management Forum (Section 4.3.1). An information model was chosen as opposed to the alternative of the more formal ontology based on its flexibility in defining optional elements. The proposed transformation of the contracts through the layers of abstraction was accomplished through Graph Theory Transformation. Graph transformation offers the formal rigour lacking in specification through Information models alone and is discussed in greater detail in Section 4.3.2. Further an example transformation and general specification is presented in Section 4.3.2.3.

7. *How can the existing concepts of market operation assist in responsible service?* Fundamentally the use of a marketplace allows for creation of on demand dynamic services, without which the ability to provide resilience through renegotiation is untenable. However, the requirement of resilience and the ability to create on demand services place some limitations on the allowable marketplace mechanisms, which is discussed in greater detail in Section 3.4. Additionally, while not directly bearing on responsibility in services, the use of the marketplace in conjunction with the CARMA system within the parameters of the simulation displayed some gains in utilisation for the individual providers (Section 6.4.5).

7.3 Original Contributions

The research in this thesis has presented a number of original concepts in the field of end-to-end performance management of complex services. These concepts and the details of their development are described below.

- The CARMA system - This CARMA system represents a concrete specification of an alternate approach to the management of end-to-end complex services. Drawing on previous concepts of agent design, and marketplace based service management, CARMA presents a competitive based complex service management system focusing primarily on Responsibility through resilience and risk management that attempts to work in concert with current management systems. (Section 3.5)
- Contract Transformation - The use of Graph transformation with information models presents a novel approach to the specification of Contracts at the different levels of abstraction (Section 4.3)
- The Service Information Management System - The fundamental requirement of risk management in the contraction of new services and in the performance of managed services represents a change in focus for the management of complex end-to-end services. The application of a Bayesien Belief Network in risk judgement represents a new view of non-functional service requirements in telecommunications and service management (Section 4).
- The CARMA model and simulation - The experimental approach of this thesis has

produced a robust model that represents a new way of exploring the issues surrounding the development of management systems in an area of increasing complexity (Section 5.4).

7.4 Future Work

The realm in which the CARMA system operates indicates that there is a great deal of potential for further investigation. One of the primary areas of exploration should be based on the intial assumptions on which this work is based (Section 3.6).

One assumption is that the aggregated performance information, as well as the utilisation of the resource that is needed by the Bundled Service Agent, to judge the risk of failure or contracting a new service would be available at the required level. Due to organisations security and trust concerns this is not guaranteed, though there is evidence that certain amounts of information can and will be provided. However, further research might attempt to build a more complete information model representation from a minimal data set, such as information collected directly by the Bundled Service Agent in the case of more intransigent Single Service Providers.

Concurrently, an investigation into the utilisation of data mining techniques on the collected historical performance information to provide more detailed inputs to the Bayesien Belief Network risk judgement is also recommended.

There was an acknowledgement in Section 2.7, that as yet there is no established mechanism for the automated and online migration of virtual machines across multiple heterogeneous clouds. Establishment of such a mechanism would greatly benefit the establishment of a market based competitive management structure such as CARMA. Additionally, further investigation is needed in the area of integration with the policy based autonomic management systems surveyed in the literature.

With regards to CARMA system itself, the functionality of the User Agent with regards to its ability to integrate with client systems for the provisioning of remote services is an area which requires greater exploration and specification.

Additionally the current implementation of performance management in the SIMS resulted in rather conservative judgements of poor performance. A greater exploration of alternatives to the algorithmic approach implemented in the model is advised. One approach, based on Support Vector Machine (Cauwenberghs and Poggio, 2001) has been proposed by this thesis and a greater investigation would be required.

The consequences on the market and on the Bundled Service Provider in utilising the proposed but unimplemented Market Action node of the BBN needs to be further investigated for its utility in increasing the judgement of risk of the providers and of the profitability of the Bundled Service Provider.

Recently there has been an exploration of Software Defined Networking (SDN) as a means to improve the management of individual networks (Kim and Feamster, 2013). While the area shows much promise in incorporating the fundamentals of autonomic network management at the router and switch level and is briefly mentioned in Section 2.4.3.1 the has been no exploration of the use of SDN within the structure of CARMA. However the author believes that the flexibility of SDN would be of great benefit to the entities of CARMA, specifically the Single Service Providers and the Single Service Resource Providers.

Finally there is further work to be done in the simulation. Firstly, the implementation of the Secondary market in the simulation to explore the effect of the market on profitability for the Bundled Service Providers. Secondly, the implementation of the information model contracts and graph transformations, with regards to the individual services types and the limitations placed by the specification of individual providers.

Bibliography

Abdel-Aty-Zohdy, H., Sherif, M. and Smiarowski, A. (2006). Three bio-inspired approaches to telecommunications defect-tracking and reliability-estimation, *Circuits and Systems, 2006. MWSCAS '06. 49th IEEE International Midwest Symposium on*, Vol. 2, pp. 542 –546.

Agoulmine, N. (2010). *Autonomic network management principles: from concepts to applications*, Academic Press.

Amazon (2012a). Regions and availability zones. Accessed on 15-1-2012.
URL: *http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html*

Amazon (2012b). Spot instances, *Amazon AWS* . Accessed 15-1-2012.
URL: *http://aws.amazon.com/ec2/spot-instances/*

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2010). A view of cloud computing, *Commun. ACM* **53**(4): 50–58.

Arno, R., Gross, P. and Schuerger, R. (2006). What five 9's really means and managing expectations, *Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE*, Vol. 1, pp. 270 –275.

Balasubramaniam, S., Barrett, K., Donnelly, W., van der Meer, S. and Strassner, J. (2006). Bio-inspired policy based management (biopbm) for autonomic bio-inspired policy based management (biopbm) for autonomic, *Policies for Distributed Systems and Networks, 2006. Policy 2006. Seventh IEEE International Workshop on*, pp. 3 –12.

Bashar, A., Parr, G., McClean, S. I., Scotney, B. W. and Nauck, D. (2010). Knowledge discovery using bayesian network framework for intelligent telecommunication network management, *Knowledge Science, Engineering and Management*, pp. 518–529.

Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z. and Weiss, W. (1998). An Architecture for Differentiated Services, RFC 2475 (Informational). Updated by RFC 3260.
URL: *http://www.ietf.org/rfc/rfc2475.txt*

Borissov, N. and Wirstrm, N. (2008). Q-strategy: A bidding strategy for market-based allocation of grid services, *in* R. Meersman and Z. Tari (eds), *On the Move to Meaningful Internet Systems: OTM 2008*, Vol. 5331 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 744–761.

Broberg, J., Buyya, R. and Tari, Z. (2009). Metacdn: Harnessing 'storage clouds' for high performance content delivery, *Journal of Network and Computer Applications* **32**(5): 1012 – 1022. Next Generation Content Networks.

Buyya, R., Pandey, S. and Vecchiola, C. (2009). Cloudbus toolkit for market-oriented cloud computing, *in* M. Jaatun, G. Zhao and C. Rong (eds), *Cloud Computing*, Vol. 5931 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 24–44.

Buyya, R., Ranjan, R. and Calheiros, R. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services, *in* C.-H. Hsu, L. Yang, J. Park and S.-S. Yeo (eds), *Algorithms and Architectures for Parallel Processing*, Vol. 6081 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 13–31.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F. and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Softw. Pract. Exper.* **41**(1): 23–50.

Calheiros, R. N., Vecchiola, C., Karunamoorthy, D. and Buyya, R. (2012). The aneka platform and qos-driven resource provisioning for elastic applications on hybrid clouds, *Future Generation Computer Systems* **28**(6): 861 – 870. Including Special sections SS: Volunteer Computing and Desktop Grids and SS: Mobile Ubiquitous Computing.

Case, J., Fedor, M., Schoffstall, M. and Davin, J. (1990). Simple network management protocol (snmp), RFC 1157 (Historic).

Cauwenberghs, G. and Poggio, T. (2001). Incremental and decremental support vector machine learning, *Advances in neural information processing systems* pp. 409–415.

Chang, X. (1999). Network simulations with opnet, *Proceedings of the 31st conference on Winter Simulation: Simulation-a bridge to the future - Volume 1*, WSC '99, ACM, New York, NY, USA, pp. 307–314.

Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization, *Computer Networks* **54**(5): 862 – 876.

Clark, D. D., Partridge, C., Ramming, J. C. and Wroclawski, J. T. (2003). A knowledge plane for the internet, *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, ACM, New York, NY, USA, pp. 3–10.

Costa, P., Ladeira, M., Carvalho, R., Laskey, K., Santos, L. and Matsumoto, S. (2008). A first-order bayesian tool for probabilistic ontologies, *Proceedings of the 21st Florida AI Research Symposium (FLAIRS).*

Dastjerdi, A. and Buyya, R. (2012). An autonomous reliability-aware negotiation strategy for cloud computing environments, *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, pp. 284 –291.

Davy, S., Jennings, B. and Strassner, J. (2008). The policy continuum: Policy authoring and conflict analysis, *Computer Communications* **31**(13): 2981 – 2995. Special Issue:Self-organization and self-management in communications as applied to autonomic networks.
URL: *http://www.sciencedirect.com/science/article/pii/S0140366408002302*

Denford, M., O'Neill, T. and Leaney, J. (2003). In *Architecture-Based Design of Computer Based Systems*, Vol. 0, IEEE Computer Society.

Derbel, H., Agoulmine, N. and Salan, M. (2009). Anema: Autonomic network management architecture to support self-configuration and self-optimization in ip networks, *Computer Networks* **53**(3): 418 – 430.
URL: *http://www.sciencedirect.com/science/article/B6VRG-4TW14YJ-3/2/41147806c839b89928697fc9b724d880*

Dick, B. (1999). What is action research?
URL: *http://www.scu.edu.au/schools/gcm/ar/whatisar.html*

Dick, B. (2000). Cycles within cycles.
URL: *http://www.scu.edu.au/schools/gcm/ar/arp/aandr.html*

Dick, B. (2002). Action research: action and research.
URL: *http://www.scu.edu.au/schools/gcm/ar/arp/aandr.html*

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs, *NUMERISCHE MATHEMATIK* **1**(1): 269–271.

DMTF (2011). Common information model schema. Accessed on 9-10-2011.
URL: *http://dmtf.org/standards/cim/cim_schema_v2260*

Dobson, S., Sterritt, R., Nixon, P. and Hinchey, M. (2010). Fulfilling the vision of autonomic computing, *Computer* **43**(1): 35 –41.

Ehrig, H., Ehrig, K., Prange, U. and Taentzer, G. (2006). *Fundamentals of Algebraic Graph Transformation (Monographs in Theoretical Computer Science. An EATCS Series)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Ejarque, J., Sirvent, R. and Badia, R. (2010). A multi-agent approach for semantic resource allocation, *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pp. 335 –342.

Eshuis, R. (2005). *Statecharting petri nets*, Beta, Research School for Operations Management and Logistics.

Eshuis, R. (2009). Translating safe petri nets to statecharts in a structure-preserving way, *Proceedings of the 2nd World Congress on Formal Methods*, FM '09, Springer-Verlag, Berlin, Heidelberg, pp. 239–255.

Famaey, J., Latre, S., Strassner, J. and De Turck, F. (2010). A hierarchical approach to autonomic network management, *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pp. 225 –232.

Femminella, M., Francescangeli, R., Reali, G., Lee, J. and Schulzrinne, H. (2011). An enabling platform for autonomic management of the future internet, *Network, IEEE* **25**(6): 24 –32.

Ferrer, A. J., Hernndez, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R. M., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, S. K., Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Kipp, A., Wesner, S., Corrales, M., Forg, N., Sharif, T. and Sheridan, C. (2012). Optimis: A holistic approach to cloud service provisioning, *Future Generation Computer Systems* **28**(1): 66 – 77.
URL: *http://www.sciencedirect.com/science/article/pii/S0167739X1100104X*

Forum, T. (2012). Shared information / data (sid) model. Accessed on 12-1-2012.
URL: *http://www.tmforum.org/InformationFramework/1684/Home.html*

Garg, S., Venugopal, S. and Buyya, R. (2008). A meta-scheduler with auction based resource allocation for global grids, *Parallel and Distributed Systems, 2008. ICPADS '08. 14th IEEE International Conference on*, pp. 187 –194.

Gibney, M. and Jennings, N. (1998). Dynamic resource allocation by market-based routing in telecommunications networks, *in* S. Albayrak and F. Garijo (eds), *Intelligent Agents for Telecommunication Applications*, Vol. 1437 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 102–117.

Giri, N., Mundle, S., Ray, A. and Bodhe, S. (2009). Multi agent system based service architectures for service level agreement in cellular networks, *Proceedings of the 2nd Bangalore Annual Compute Conference*, COMPUTE '09, ACM, New York, NY, USA, pp. 5:1–5:6.

Google (2012). App engine. Accessed on 15-1-2012.
URL: *https://developers.google.com/appengine/docs/python/datastore/ structuring_for_strong_consistency*

Goscinski, A. and Brock, M. (2010). Toward dynamic and attribute based publication, discovery and selection for cloud computing, *Future Generation Computer Systems* **26**(7): 947 – 970.

Green, L. (2004). Service level negotiation in a heterogeneous telecommunications environment, *Proceeding International Conference on Computing, Communications and Control Technologies (CCCT04)*, Austin, TX, USA.

Hall, R. E. (2001). *Digital dealing: how e-markets are transforming the economy*, W. W. Norton & Company, Inc., 500 Fifth Avenue, New York, NY, 10110.

Hao, F., Lakshman, T. V., Mukherjee, S. and Song, H. (2009). Enhancing dynamic cloud-based services using network virtualization, *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, VISA '09, ACM, New York, NY, USA, pp. 37–44.

Harel, D. (1987). Statecharts: A visual formalism for complex systems, *Science of computer programming* **8**(3): 231–274.

Heckel, R., Cherchago, A. and Lohmann, M. (2004). A formal approach to service specification and matching based on graph transformation, *Electronic Notes in Theoretical Computer Science* **105**(0): 37 – 49. Proceedings of the First International Workshop on Web Services and Formal Methods (WSFM 2004).
URL: *http://www.sciencedirect.com/science/article/pii/S1571066104051588*

Hedayati, A., Fesharaki, M., Badie, K. and Aghazarian, V. (2011). Arishtat: auction-based dynamic bandwidth allocation method in ethernet passive optical networks, *Communications, IET* **5**(15): 2116 –2124.

Hewlett-Packard Development Company, L. (2009). Hp network automation software version: 7.60 user guide, http://h20230.www2.hp.com/selfsolve/manuals.

Inoue, Y., Cuha, D. and Berndt, H. (1998). The tina consortium, *Communications Magazine, IEEE* **36**(9): 130 –136.

Jennings, B., Brennan, R., Donnelly, W., Foley, S., Lewis, D., O'Sullivan, D., Strassner, J. and van der Meer, S. (2009). Challenges for federated, autonomic network management in the future internet, *Integrated Network Management-Workshops, 2009. IM '09. IFIP/IEEE International Symposium on*, pp. 87 –92.

Jennings, B., van der Meer, S., Balasubramaniam, S., Botvich, D., Foghlu, M., Donnelly, W. and Strassner, J. (2007). Towards autonomic management of communications networks, *Communications Magazine, IEEE* **45**(10): 112 –121.

Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri Nets Modelling and Validation of Concurrent Systems*, Springer.

Johnson, B. (2008). The Guardian Newspaper.

Josephsen, D. (2007). *Building a Monitoring Infrastructure with Nagios*, Prentice Hall PTR, Upper Saddle River, NJ, USA.

Kahani, M. and Beadle, H. W. P. (1997). Decentralised approaches for network management, *SIGCOMM Comput. Commun. Rev.* **27**(3): 36–47.
URL: *http://doi.acm.org/10.1145/263932.263940*

Kant, U. and Grosu, D. (2005). Double auction protocols for resource allocation in grids, *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, Vol. 1, pp. 366 – 371 Vol. 1.

Kemmis, S. and McTaggart, R. (1988). *The action research planner*, Deakin University.

Kephart, J. and Chess, D. (2003). The vision of autonomic computing, *Computer* **36**(1): 41–50.

Kim, H., Benson, T., Akella, A. and Feamster, N. (2011). The evolution of network configuration: a tale of two campuses, *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ACM, pp. 499–514.

Kim, H. and Feamster, N. (2013). Improving network management with software defined networking, *Communications Magazine, IEEE* **51**(2): 114–119.

Kim, H. and Parashar, M. (2011). Cometcloud: An autonomic cloud engine, *Cloud Computing: Principles and Paradigms* pp. 275–297.

Koch, T., Krell, C. and Kraemer, B. (1996). Policy definition language for automated management of distributed systems, *Proceedings of the 2nd IEEE International Workshop on Systems Management (SMW'96)*, SMW '96, IEEE Computer Society, Washington, DC, USA, pp. 55–.

LaBarre, L. (1991). Osi internet management: Management information base, RFC 1214 (Historic).

Lee, J. W., Francescangeli, R., Janak, J., Srinivasan, S., Baset, S., Schulzrinne, H., Despotovic, Z. and Kellerer, W. (2011). Netserv: Active networking 2.0, *Communications Workshops (ICC), 2011 IEEE International Conference on*, pp. 1 –6.

Lee, M., Ye, X., Marconett, D., Johnson, S., Vemuri, R. and Ben Yoo, S. (2008). Autonomous network management using cooperative learning for network-wide load balancing in heterogeneous networks, *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1 –5.

Lefevre, L. (2005). Heavy and lightweight dynamic network services : challenges and experiments for designing intelligent solutions in evolvable next generation networks, pp. 738 – 743.

Li, Z., Sim, C. H. and Low, M. (2006). A survey of emergent behavior and its impacts in agent-based systems, *Industrial Informatics, 2006 IEEE International Conference on*, pp. 1295 –1300.

Lorimer, T. and Sterritt, R. (2012). Autonomic management of cloud neighbourhoods through pulse monitoring.

Lymberopoulos, L., Lupu, E. and Sloman, M. (2003). An adaptive policy-based framework for network services management, *Journal of Network and Systems Management* **11**: 277–303. 10.1023/A:1025719407427.
URL: *http://dx.doi.org/10.1023/A:1025719407427*

Maille, P. and Tuffin, B. (2006). Pricing the internet with multibid auctions, *Networking, IEEE/ACM Transactions on* **14**(5): 992 –1004.

Maxwell, C. (2007). *Representing heuristics for architectural optimisation*, PhD thesis, University of Technology, Sydney.

Mearns, H., Leaney, J. and Verchere, D. (2010a). The architectural evolution and performance bottlenecks of telecommunications network management systems, *Proceedings of the 17th IEEE International Conference and Workshops on Engineering of Computer-Based Systems*, IEEE, pp. 281–285.

Mearns, H., Leaney, J. and Verchere, D. (2010b). Critique of network management systems and their practicality, *Engineering of Autonomic and Autonomous Systems (EASe), 2010 Seventh IEEE International Conference and Workshops on*, pp. 51 –59.

Microsoft (2012). Azure. Accessed on 15-1-2012.
URL: *http://www.windowsazure.com/en-us/develop/net/best-practices/performance/*

Ming, L., Hong, J. and Xi, L. (2009). Vcg auction based hierarchical bandwidth sharing scheme for integrated heterogeneous networks, *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, pp. 1 –4.

Mykoniati, E., Charalampous, C., Georgatsos, P., Damilatis, T., Goderis, D., Trimintzios, P., Pavlou, G. and Griffin, D. (2003). Admission control for providing qos in diffserv ip networks: the tequila approach, *Communications Magazine, IEEE* **41**(1): 38 – 44.

Na, W. and Ping, W. (2010). Quality of learning analysis based on bayesian network, *Computer Design and Applications (ICCDA), 2010 International Conference on*, Vol. 5, pp. V5–382 –V5–384.

Newman, M. E. J. (2005). In *Random graphs as models of networks*, Wiley-VCH Verlag GmbH & Co. KGaA.

Nimis, J., Anandasivam, A., Borissov, N., Smith, G., Neumann, D., Wirstrm, N., Rosenberg, E. and Villa, M. (2008). Sorma ? business cases for an open grid market: Concept and implementation, *in* J. Altmann, D. Neumann and T. Fahringer (eds), *Grid Economics and Business Models*, Vol. 5206 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 173–184.

OWL Web Ontology Language (2010). Source: http://www.w3.org/TR/2004/REC-owlguide- 20040210/.

Pourebrahimi, B., Ostadzadeh, S. and Bertels, K. (2008). Resource allocation in market-based grids using a history-based pricing mechanism, *in* T. Sobh (ed.), *Advances in Computer and Information Sciences and Engineering*, Springer Netherlands, pp. 97–100.

Ranjan, R., Buyya, R. and Parashar, M. (2012). Special section on autonomic cloud computing: technologies, services, and applications, *Concurrency and Computation: Practice and Experience* **24**(9): 935–937.
URL: *http://dx.doi.org/10.1002/cpe.1865*

Reddy, S. and Gupta, A. (2006). Auction based resource allocation in grids, *in* S. Chaudhuri, S. Das, H. Paul and S. Tirthapura (eds), *Distributed Computing and Networking*, Vol. 4308 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 145–156.

Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., Montero, R., Wolfsthal, Y., Elmroth, E., Caceres, J., Ben-Yehuda, M., Emmerich, W. and Galan, F. (2009). The reservoir model and architecture for open federated cloud computing, *IBM Journal of Research and Development* **53**(4): 4:1 –4:11.

Rosen, E., Viswanathan, A. and Callon, R. (2001). Multiprotocol Label Switching Architecture, RFC 3031 (Proposed Standard). Updated by RFC 6178.
URL: *http://www.ietf.org/rfc/rfc3031.txt*

Rubio-Montero, A., Huedo, E., Montero, R. and Llorente, I. (2007). Management of virtual machines on globus grids using gridway, *Parallel and Distributed Processing Symposium, International* **0**: 358.

Rumbaugh, J., Jacobson, I. and Booch, G. (2004). *Unified Modeling Language Reference Manual, The (2nd Edition)*, Pearson Higher Education.

Schön, D. (1983). *The Reflective Practitioner: How Professionals Think in Action*, Basic Books.

Semret, N., Liao, R.-F., Campbell, A. and Lazar, A. (2000). Pricing, provisioning and peering: dynamic markets for differentiated internet services and implications for network interconnections, *Selected Areas in Communications, IEEE Journal on* **18**(12): 2499 –2513.

Serrano, M., van Der Meer, S., Holum, V., Murphy, J. and Strassner, J. (2010). Federation, a matter of autonomic management in the future internet, *Network Operations and Management Symposium (NOMS), 2010 IEEE*.

Sheers, K. R. (1996). Hp openview event correlation services, *Hewlett Packard Journal* **47**: 31–33.

Sheridan-Smith, N. (2007). *A Distributed Policy-based Management (PBM) System for Complex Networks and Services*, PhD thesis, University of Technology.

Sheridan-Smith, N., O'Neill, T., Leaney, J. and Hunter, M. (2006). A policy-based service definition language for service management, *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pp. 282 –293.

Song, S., Hwang, K. and Kwok, Y.-K. (2006). Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling, *Computers, IEEE Transactions on* **55**(6): 703 –719.

Stantchev, V. and Schrpfer, C. (2009). Negotiating and enforcing qos and slas in grid and cloud computing, *in* N. Abdennadher and D. Petcu (eds), *Advances in Grid and Pervasive Computing*, Vol. 5529 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 25–35.

Steinder, M. and Sethi, A. S. (2004). A survey of fault localization techniques in computer networks, *Science of Computer Programming* **53**(2): 165 – 194.
URL: *http://www.sciencedirect.com/science/article/pii/S0167642304000772*

Stenzhorn, H., Schulz, S., Boeker, M. and Smith, B. (2008). Adapting clinical ontologies in real-world environments., *Journal of universal computer science : J. UCS* **14**(22).
URL: *http://europepmc.org/abstract/MED/20390048*

Strassner, J. (2003). *Policy-Based Network Management: Solutions for the Next Generation (The Morgan Kaufmann Series in Networking)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Strassner, J. (2009). Context-aware, policy-based seamless mobility using the focale autonomic architecture, *Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on*, pp. 568 –573.

Taentzer, G. (2004). Agg: A graph transformation environment for modeling and validation of software, *in* J. Pfaltz, M. Nagl and B. Bhlen (eds), *Applications of Graph Transformations with Industrial Relevance*, Vol. 3062 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 446–453.

TMForum (2010). Bundled service definition.
URL: *http://www.tmforum.org/*

Turner, D. M., Prevelakis, V. and Keromytis, A. D. (2010). A market-based bandwidth charging framework, Vol. 10, ACM, New York, NY, USA, pp. 1:1–1:30.

Tusa, F., Celesti, A., Paone, M., Villari, M. and Puliafito, A. (2011). How clever-based clouds conceive horizontal and vertical federations, *Computers and Communications (ISCC), 2011 IEEE Symposium on*, pp. 167 –172.

Tusa, F., Paone, M., Villari, M. and Puliafito, A. (2010). Clever: A cloud-enabled virtual environment, *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pp. 477 –482.

van der Aalst, W. M. P. (1998). The application of petri nets to workflow management., *Journal of Circuits, Systems, and Computers* **8**(1): 21–66.

Vaughan-Nichols, S. (2011). Openflow: The next generation of the network?, *Computer* **44**(8): 13 –15.

Venugopal, S., Chu, X. and Buyya, R. (2008). A negotiation mechanism for advance resource reservations using the alternate offers protocol, *Quality of Service, 2008. IWQoS 2008. 16th International Workshop on*, pp. 40 –49.

Voruganti, R. (1994). A global network management framework for the 90s, *Communications, 1994. ICC '94, SUPERCOMM/ICC '94, Conference Record, 'Serving Humanity Through Communications.' IEEE International Conference on*, pp. 1094–1098 vol.2.

Wallin, S. and Leijon, V. (2009). Telecom network and service management: An operator survey, *in* T. Pfeifer and P. Bellavista (eds), *Wired-Wireless Multimedia Networks and Services Management*, Vol. 5842 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 15–26.

Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*, 2nd edn, Wiley Publishing.

XJTechnologies (2010). Anylogic 6 educational, http://www.xjtek.com.

Yan, S., Lee, B. S., Zhao, G., Ma, D. and Mohamed, P. (2011). Infrastructure management of hybrid cloud for enterprise users, *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on*, pp. 1 –6.

Yin, R. K. (2003). *Case study research: design and methods*, 3rd edn, Sage Publications.

Appendix A

Refereed Publications

During the course of the research presented in this thesis a number of refereed publications have been produced to present the developed ideas. For the reader's convenience, these papers have been provided in this Appendix.

[Production Note: the papers are not included in this digital copy due to copyright restrictions. The print copy includes the fulltext of the papers and can be viewed at UTS Library]

Mearns, H., Leaney, J. and Verchère, D. (2010). The architectural evolution of telecommunications network management systems, in proceeding of *17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, ECBS 2010, Oxford, England, UK, 22-26 March 2010, pp. 281-285. DOI: 10.1109/ECBS.2010.39.
View/Download from: Publisher's site

Mearns, H., Leaney, J. and Verchère, D. (2010). Critique of network management systems and their practicality, in proceeding of *17th IEEE International Conference and Workshops on the Engineering of Autonomic and Autonomous Systems*, ECBS 2010, Oxford, England, UK, 22-26 March 2010, pp. 51-59. DOI: 10.1109/EASe.2010.15.
View/Download from: Publisher's site

Mearns, H., Leaney, J., Parakhine, A., Debenham, J.K. and Verchère, D. (2011). An autonomic open marketplace for service management and resilience, *CNSM*, IEEE, pp. 1-5.
View/Download from: Publisher's site

Mearns, H., Leaney, J., Parakhine, A., Debenham, J.K. and Verchère, D. (2011). An autonomic open marketplace for inter-cloud service management, in proceeding of *4th IEEE International Conference on Utility and Cloud Computing*, UCC 2011, Melbourne, Australia, 5-8 December 2011, pp. 186-193. DOI: 10.1109/UCC.2011.34
View/Download from: Publisher’s site

Mearns, H., Leaney, J., Parakhine, A., Debenham, J.K. and Verchère, D. (2012). CARMA: complete autonomous responsible management agents for telecommunications and inter-cloud services, *IEEE Network Operations and Management Symposium: Mini-Conference*, NOMS 2012, pp. 1089-1095.
View/Download from: Publisher’s site