Incorporating an Element of Negotiation into a Service-Oriented Broker Application

PINTO, PONGPAN

Incorporating an Element of Negotiation into a Service-Oriented Broker Application

Pongpan Pinto

A thesis submitted to the Durham University for the degree of Doctor of Philosophy

Department of Computer Science

School of Engineering and Computing Sciences

Durham University

2010

Abstract

The Software as a Service (SaaS) model is a service-based model in which a desired service is assembled, delivered and consumed on demand. The IBHIS broker is a 'proof of concept' demonstration of SaaS which is based on services that deliver data. IBHIS has addressed a number of challenges for several aspects of service-based software, especially the concept of a 'broker service' and service negotiation that is only used in establishing end-user access authorizations.

This thesis investigates and develops an extended form of service-based broker, called CAPTAIN (Care Planning Through Auction-based Information Negotiation). It extends the concepts and role of the broker as used in IBHIS, and in particular, it extends the service negotiation function in order to demonstrate a full range of service characteristics. CAPTAIN uses the idea of the integrated care plan from healthcare to provide a case study. A care planner acting on behalf of a patient uses the broker to negotiate with providers to produce the integrated care plan for the patient with the broker and the providers agreeing on the terms and conditions relating to the supply of the services.

We have developed a 'proof of concept' service-oriented broker architecture for CAPTAIN that includes planning, negotiation and service-based software models to provide a flexible care planning system. The CAPTAIN application has been evaluated that focuses on three features: functions, data access and negotiation. The CAPTAIN broker performs as planned, to produce the integrated care plan. The providers' data sources are accessed to read and write data records during and after service negotiation. The negotiation model permits the broker to interact with the providers to produce an adaptable plan, based on the client's needs. The primary outcome is an extendable service-oriented broker architecture that can enable more scalable and flexible distributed information management by adding interaction with the data sources.

Contents

1 Introduction **1**

1.1 Statement of Problem 1

1.1.1 Information Gathering 5

1.1.2 Planning 8

1.1.3 Research Question 11

1.2 Research Method 13

1.3 The Criteria for Success 18

1.4 Contributions 20

1.5 Overview of the Thesis 23

2 Service-based Software **25**

2.1 The Service Concept in Software 25

2.2 Service-Based Software 28

2.3 Software as a Service 31

2.4 SaaS and SOA 33

2.4.1 Service-Oriented Architecture (SOA) 33

2.4.2 Comparison Between SOA and SaaS 36

2.5 Web Services 39

2.6 Summary 41

3 The IBHIS Broker **42**

3.1 Introduction 42

3.2 The Broker Concept 43

3.3 The Architecture of IBHIS 44

3.4 The Data Access Service (DAS) 47

3.5 Operation of IBHIS 49

3.6 Services in IBHIS 51

3.7 Implementation 51

3.8 Summary 55

4 Negotiation **56**

4.1 Negotiation Fundamentals 56

4.1.1 The Nature of Negotiation 57

4.1.2 Distributive and Integrative Negotiations 59

4.2 Types of Negotiation 60

4.2.1 Auction 60

4.2.2 Game Theory 64

4.2.3 Bargaining 65

4.3 Automated Negotiation 66

4.3.1 Service-Oriented Architecture 68

4.3.2 Electronic Commerce 70

4.3.3 Web Services 74

4.3.4 Artificial Intelligence 76

4.4 Summary 77

5 The Integrated Care Plan **79**

5.1 Introduction 79

5.2 The Nature of the Integrated Care Plan 81

5.2.1 Healthcare in the UK 82

5.2.2 Case Management and Care Planning 85

5.2.3 The Integrated Care Plan and a Care Planning Broker 87

5.3 The Care Planning System 89

5.3.1 The Integrated Care Plan 89

5.3.2 The Conceptual Framework 90

5.3.3 System Operation 93

5.4 An Example Care Plan 95

5.4.1 Use Case 95

5.4.2 Scenarios 96

5.4.2.1 Scenario: Create a Care Plan 96

5.4.2.2 Scenario: Modify the Care Plan 97

5.5 Summary 99

6 The CAPTAIN Broker 101

6.1 Introduction 101

6.2 The Architecture of CAPTAIN 105

6.3 Negotiation Model 109

6.3.1 The Negotiation Process 111

6.3.2 The Negotiation Object 112

6.3.3 The Negotiation Protocol 113

6.3.4 The Decision Model 115

6.3.4.1 The P-Broker 115

6.3.4.2 The DASs 115

6.4 Service Negotiation in CAPTAIN 117

6.4.1 Pre-negotiation 118

6.4.2 Service Negotiation 119

6.4.3 Post-negotiation 121

6.5 Implementation 122

6.5.1 The Original Prototype 122

6.5.2 The CAPTAIN Application 123

6.5.3 Development and Deployment Tools 125

6.6 Summary 127

7 Evaluation 129

7.1 Introduction 129

7.2 Scenarios 132

7.2.1 Creation of a Care Plan 132

7.2.2 Modification of the Care Plan 136

7.2.3 Management of Appointment Conflict 138

7.2.4 Management of Parallel Access and Timeout 141

7.2.5 The Role of Read and Write Data Sources 144

7.2.6 Timeout of Reservation and Booking Services 146

7.2.7 Proposition and Selection Offers of the DASs and the P-Broker 149

7.2.8 System Observation 152

7.2.8.1 Ordering of Events in Distributed Systems 153

7.2.8.2 Ordering of Events in Service Negotiation for CAPTAIN 154

7.3 Summary 156

8 Discussion 157

8.1 Achievements 157

8.2 Limitations 161

8.3 Addressing the Research Question 165

8.4 Summary 168

9 Conclusions 170

9.1 Summary 170

9.2 Further Work 175

List of Figures

1.1 Distributed information management 3

1.2 Distributed information management with interaction 4

1.3 The conceptual framework of planning service system 10

1.4 Research framework 16

1.5 The framework of the concept implementation 21

2.1 Layered service model taken from (Turner et al., 2003) 31

2.2 Service Oriented Architecture taken from (Erl, 2007) 34

2.3 The key abstraction of an SOA taken from (Krafzig et al., 2004) . . 35

2.4 The overlapping concepts of SaaS and SOA taken from (Singla, 2009) 37

3.1 A schematic of the IBHIS structure taken from (Budgen et al., 2005) 44

3.2 The architecture of IBHIS taken from(Rigby et al., 2007) 45

3.3 The architecture of DAS taken from (Budgen et al., 2005) 47

3.4 The processing a query in IBHIS taken from (Rigby *et al.*, 2007) . . 50

3.5 IBHIS role/team activation screen 53

3.6 Ontological query formulation screen 54

3.7 The result from IBHIS's query 54

4.1 Negotiation research areas, their results and key influences taken from (Bichler et al., 2003) 57

4.2 Market framework taken from (Teich et al., 1999) 61

4.3 The agents' bargaining protocol taken from (Gerding et al., 2006) . 67

4.4 The three phases of negotiating in a service-oriented environment taken from (Elfatatry and Layzell, 2004) 69

4.5 The components in the negotiation framework and relationships between them taken from (Kim and Segev, 2003) 71

4.6 Key concepts in the Decision Model taken from (Li, 2001) 73

4.7 The structure of WS-Negotiation taken from (Hung et al., 2004) . . 75

4.8 Relationship of negotiation entities taken from (Hung et al., 2004) . 76

5.1 The structure of NHS taken from (NHS, 2010) 84

5.2 The negotiation framework for creating the integrated care plan . . . 89

5.3 The conceptual framework 90

5.4 The relationship between care and actual plan 92

5.5 System operation for the integrated care plan 93

6.1 The structure of CAPTAIN 103

6.2 The architecture of CAPTAIN 106

6.3 The negotiation model of CAPTAIN 111

6.4 Example of a negotiation object for a query 113

6.5 Example of a negotiation object for an offer 113

6.6 The negotiation protocol of CAPTAIN 114

6.7 The reservation states of a DASs' diary 116

6.8 The sequence diagram of CAPTAIN 118

6.9 Processing queries in the DASs 120

6.10 Processing offers in the P-Broker 121

6.11 Choosing an operation of the CAPTAIN application 123

6.12 Generating a request of the CAPTAIN application 124

6.13 Choosing offers of the CAPTAIN application 125

7.1 The ordering of events in service negotiation for CAPTAIN 155

List of Tables

1.1 Research context 12

1.2 The contribution of CAPTAIN 20

2.1 Focused Zachman model for comparing SOA and SaaS taken from (Laplante et al., 2008) 36

4.1 Characteristics of negotiations and auctions taken from (Bichler et al., 2003) 63

5.1 An example of a care plan request 98

5.2 An example of an actual plan 99

6.1 The DASs' time-out 116

7.1 Scenarios for evaluation of CAPTAIN's main features 130

7.2 The example of a request for creating a new care plan 133

7.3 An example result of creating the new care plan 135

7.4 The example of request for modifying the care plan 137

7.5 The example of result of modifying the care plan 137

7.6 The example of appointments before managing appointment conflict 140

7.7 The example of appointments after managing appointment conflict . 140

7.8 The example of appointments before managing appointment conflict 143

7.9 The example of appointments after managing appointment conflict . 143

7.10 Reserve DAS's diary 145

7.11 Booking DAS's diary 145

7.12 The example of diary slots before time-out 148

7.13 The example of diary slots after time-out 148

7.14 The example of DASs' offers 150

Declaration

The material presented in this thesis is the sole work of the author and has not been previously submitted for a degree at this or any other university.

Statement of Copyright

Acknowledgements

I would like to dedicate this work to my late father, Prom Pinto, who passed away during my study in the UK. I deeply appreciate my mother, Piengjai Pinto, for her patience and perseverance during the great lost to relieve me from worry and could concentrate on my study.

In 2005, I moved from Keele University to Durham University following Professor David Budgen with trust and hope. I would like to express my sincere gratitude and appreciation to him for his guidance and constructive advise as well as his encouragement throughout my study. He played a perfect role as a supervisor and this will be my model for my professor life.

I am grateful to Professor Malcolm Munro and Professor Pearl Brereton for their advice and guidance. I also thank Professor Michael Rigby who supported me with much information for my research. Special thanks go to Dr. Mark Turner who had helped me, particulary in the subject of system implementation. I appreciate the examiners, Professor Paul Layzell and Dr. William Wei Song, for their suggestion and final wrap up.

I treasure the two boys, Tinh and Ziewfong, for their hopes and looking forwards to see me back to Thailand. They were always beside and supported me, especially during my most difficult time. Their innocent words gave me a lot more energy to move forward. I thank my brother and sister who had taken care of my mother very well during my study.

I acknowledge the financial support to the Royal Thai government and Thaksin University. I also express my appreciation to Faculty of Science, Thaksin University for granting me a study leave.

Chapter 1

Introduction

This chapter first presents the motivation for this research and then describes the research methodology adopted for the investigation and the development of a service oriented broker system incorporating elements of negotiation. It also discusses the criteria for success, identifies the author's contributions, and provides an outline of the thesis.

1.1 Statement of Problem

The conventional development of software has focused on supply-side issues, driven by developers or technology rather than end users (Budgen *et al.*, 2004). However, software developed and delivered as a product is not adaptable to the new era of rapidly changing business needs. Moreover, the internet has been increasingly used as a disruptive platform for new various kinds of business model (Domingue *et al.*, 2009). As a result, the software industry as established by software vendors has begun to move from software products to more profitable software services (IBM-SaaS, 2010). Accordingly, the new approach of software development has employed a demand-led strategy that focuses more on the needs of the end users or service consumers (Kohlborn *et al.*, 2009; Budgen *et al.*, 2004).

A further benefit of the "service-oriented" approach is that it is independent from particular software programming languages or operating systems (Papazoglou et al., 2007). The "service-oriented" approach allows software to be composed by discovering and invoking a set of services through a network of services. Hence, business services can be exposed and offered to facilitate demand from other service software or consumers.

Therefore, a service-oriented model of software can be used to develop software providing a software service that is consumed on demand and may be discarded later by the end users (Budgen *et al.*, 2004; Elfatatry & Layzell, 2004; Demirkan *et al.*, 2008). Software service-based models have been used as the means to develop software around a demand-centric aspect that leads to software delivered as a service (Bennett *et al.*, 2000; Kohlborn *et al.*, 2009). "The most significant benefit of the service model is the ability to loosely couple dynamic services, finding and binding the services as needed" (Budgen *et al.*, 2007). An example of the evolving nature of software service-oriented models is a service-oriented architecture (SOA) (Laplante *et al.*, 2008; Papazoglou *et al.*, 2007). The SOA is used to design and develop an agile service-oriented software, providing services that are dynamically and loosely coupled and that are consumed by other distributed applications or services.

A software service is "the delivery of software functionality by autonomous, distributed computation elements, usually across a suitably high-bandwidth network" (Budgen *et al.*, 2004). The software service can be essentially characterized as being used rather than owned, being independent of programming language construction, being stateless, and providing a contractual interface (Budgen *et al.*, 2007). It exhibits a set of reusable application functionalities that can be composed in regard to business requirements (Kohlborn, Korthaus, Chan and Rosemann, 2009).

The use of the software as a service model can lead to highly flexible and agile software that should be able to achieve the needs of emergent open businesses. In addition, the platform-independent nature of software services provides a flexible means of creating distributed systems, especially those that involve distributed information management, drawing information together from independent agencies.

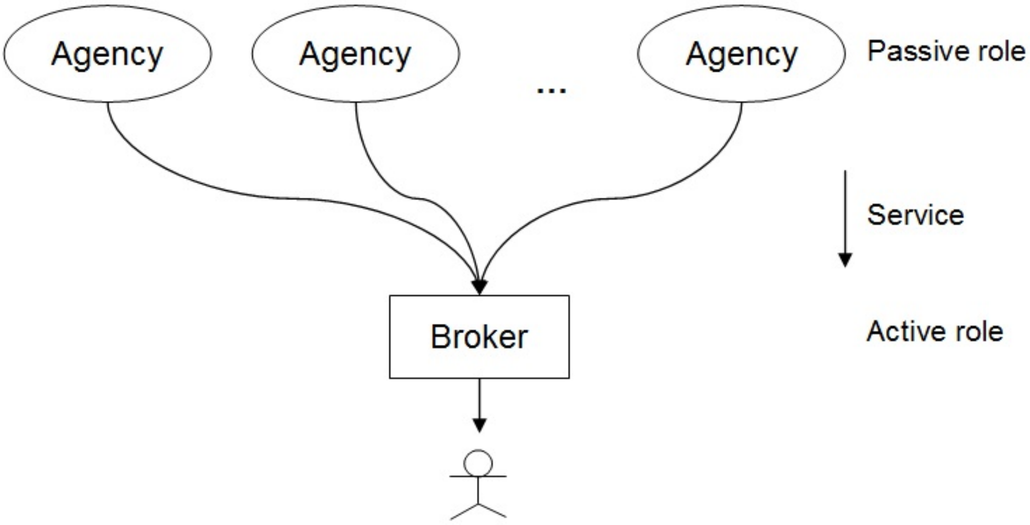


Figure 1.1: Distributed information management

Figure 1.1 shows the conceptual framework for a distributed information management strategy that involves the aggregation and management of information from a number of various sources. These sources are physically distributed and are owned by independent agencies, holding data in diverse forms. The agencies perform a passive role in the provision of information to the end users that the agencies' data access services are based on the 'read-only' property. The data sources of the agencies are accessed without the change of the content in the data sources. In contrast, the broker service plays an active role that involves identifying which sources of information should be used to address a specific query at the time when the query is relayed to the sources. The broker acts as a 'trusted intermediary' and on behalf of the end user to acquire information from the various agencies and then to manage the acquired information to meet the needs of the end user.

An example of this kind of the distributed information management system is travel services. The travel agencies act as an information broker providing services about the journey according to the queries of the end users. An agency's broker will aggregate information from a number of diverse travel providers and then manage the aggregated information into a form based on the interest of the end users.

IBHIS (Integration Broker for Heterogeneous Information Sources) is an example system of a service-based broker dealing with distributed information management (Budgen *et al.*, 2007). IBHIS has shown that distributed information management can be organised around simple information services, involving limited negotiation of data access, based upon a user's role. IBHIS uses the concept of a broker to aggregate information from the various agencies and then to blend and deliver the aggregated information to the end user.

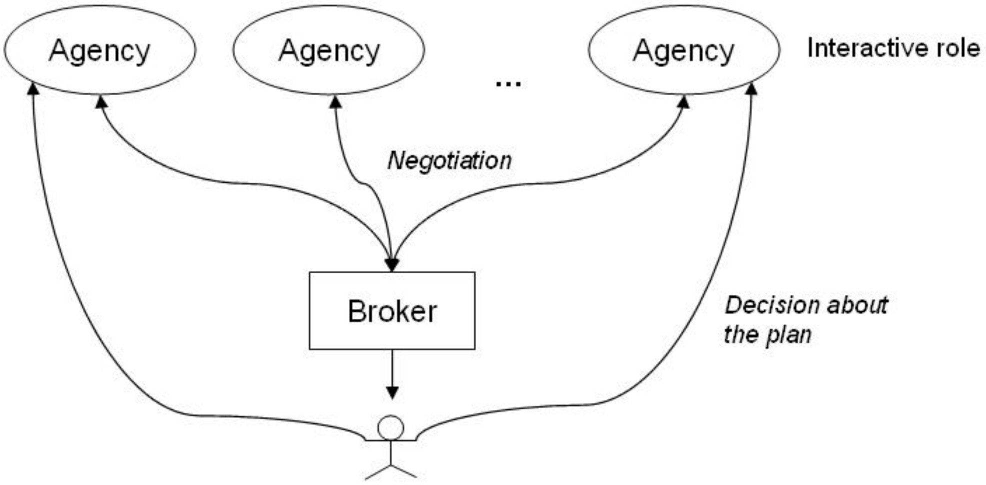


Figure 1.2: Distributed information management with interaction

However, as shown in Figure 1.2, when going beyond a passive role as a source of information, distributed information management may need some degree of interaction between the broker and the agencies in order to achieve a specific task for the end user. In such a context, the agencies perform in an interactive role that the agencies' data access services are based on the 'read-write' property. The data sources of the agencies are accessed to read and write the content in their data sources. Therefore, the agencies can interact or respond to the broker through some specific mechanisms, for instance negotiation for a plan. In such a situation, the broker acts on behalf of the end user. The broker gathers and uses the needs and preferences of the end user to negotiate with the agencies on the terms and conditions of the provision of the agencies' services in order to accomplish agreement and to resolve any conflicts of interest between them. As the result of the negotiation, the end user can make an appropriate decision about the plan.

This project has therefore been motivated by the need to develop a service-based software model that can deliver the distributed information management with interaction. The project aims to extend the concept of IBHIS to a situation where there are multiple service providers for a given function or capability and where we then want to negotiate the terms and conditions for choosing and using a given provider's service.

The following sections will describe the planning and information gathering contexts. They are the key contexts of distributed information management, both with and without interaction. Both contexts are described, together with their problems, roles and limitations, and in turn these lead to the research question.

1.1.1 Information Gathering

End users or clients have been key elements in the new era of information and technology (Budgen *et al.*, 2007). These clients, ranging across individuals, businesses, and government agencies, widely retain information in computer systems. At the same time, network technology with enhanced speed and bandwidth has expanded the degree of interconnectivity among computer systems. Consequently, the clients have an increasing demand for information while technology has an increasing capability to gather and integrate information from a number of diverse resources.

Within a distributed context, information gathering relies on the discovery of appropriate resources by dynamically finding and locating all resources that can potentially provide information that is relevant to the needs of the client. The resources are potentially distributed, heterogeneous in form and owned by autonomous information service agencies. All relevant resources providing information required by the client are first located dynamically. Then these resources are accessed to gather information for the client.

In addition, the resources may well be autonomous and employ specific rules, for instance about content authorisation and access control. These rules may require confidentiality or place restrictions upon accessing and using the resources' data. The client then has to be authorised to access the resources' data. Authorisation to the data is constrained by characteristics related to, such as, individuals, their roles and any team membership. The client therefore has to negotiate with the resources for the right of data provision. However, the role of negotiation within the context of information gathering is mainly limited to access control needed to use the data. So there is no form of *negotiation* between the client and the resources arising from possible conflicts of interest, for instance the price of a product. In such a situation, the client tries to negotiate with the resources for the cheapest price while the resources seek to obtain the highest price for the highest benefit. The result of the negotiation is a mutual price agreed by both negotiation participants.

Software as a Service (SaaS) is a software service-based model used to deal with information gathering or acquisition from distributed, heterogenous and autonomous resources dynamically (Budgen *et al.*, 2007, 2004). SaaS is a dynamic service-based approach that is based on a demand-led concept, in which desired services are assembled and used on demand to deal with a specific requirement. SaaS is organised around five main service functions: description, discovery, negotiation, delivery and composition. Service description provides the information about the services provided. It includes the service descriptions of such characteristics as functionality, non-functional aspects and constraints. The service description is used by the service delivery to locate appropriate services, resulting in a group of candidate services and providers. The service composition, then, may take place to create a new service composed from a set of lower level services. Service negotiation is a process in which the client and service providers interact to reach an agreement on the terms and conditions for the supply of a service. Service delivery is the process of actually delivering an agreed service. Therefore, SaaS is employed as a software delivery model for delivering the software service system as a service to a client. (Laplante et al., 2008; Budgen et al., 2004). The advantage of delivering software services with this approach is in reducing the limitations of software use, deployment and evolution, for instance cost and time.

IBHIS is a 'proof of concept' of SaaS demonstrating a service-based broker architecture which is based on services that deliver data (Budgen *et al.*, 2005, 2004). IBHIS is a truely service-based system. It demonstrates how potentially enormous-scale healthcare data from various data sources that are distributed, heterogeneous and autonomous can be integrated. It uses an information broker model to provide information services for the client (the end user). The information broker service "offers the opportunity of integrating data from a range of autonomous agencies, while at the same time preserving any restrictions on access to or use of the information that either agencies or legal and ethical frameworks might impose" (Budgen *et al.*, 2007). For IBHIS, the information broker acts on behalf of the client. It blends the demands and preferences of the client. Then it uses this to gather information from serveral distributed data sources and melds the gathered information into an appropriate form that meet the client's needs.

In terms of use of negotiation, IBHIS employs an element of service negotiation to determine access permissions of the end user to data services (Budgen *et al.*, 2004). However, the form of service negotiation involved does not deal with any conflict of interests between the client and service providers to reach a mutually acceptable agreement on such issues as cost. In particular, the 'service broker' aspect of IBHIS is used to provide services that exhibit only 'read-only' properties in terms of negotiation. The service access involved in the specific domain of application being 'read-only', involves making a copy of the requested data and leaving the data sources unchanged.

1.1.2 Planning

Planning is the process of setting goals, developing strategies, outlining activities and scheduling of a plan in order to accomplish the objective of a plan (Kodner and Spreeuwenberg, 2002; Kerzner, 2003; Neies and Berman, 2004). A planner or planning team works by exchanging information with a service provider and negotiating appropriate terms and conditions in order to produce and manage the plan. Then a number of participants perform or operate a set of actions or activities according to the plan.

An example of planning is the process of organising a mortgage loan for a home buyer. A mortgage broker creates a mortgage plan for the home buyer. The broker finds and negotiates with potential lenders for the best deal for a loan that will meet the needs of the home buyer. Therefore, negotiation is an important process for developing the plan that meets the needs of a service consumer as well as the constraints of the service providers.

Negotiation is "a form of interaction aimed at tailoring needs dynamically" (Elfatatry and Layzell, 2004), especially within intensely dynamic service environments where consumers and providers are continually forced to change their demands and the supply of services respectively. Negotiation involves the process of 'give-and-take' that is the "heart of negotiation" (Lewicki et al., 2006). Negotiation participants use negotiation as a decision-making approach in order to seek for mutual agreements according to the conflicts of interest between them (Bichler *et al.*, 2003). The negotiation participants can negotiate with each other based on both functional and non-functional issues, such as cost and quality of service (Dang and Huhns, 2005). There are many types of negotiation protocols, such as an auction. The auction is a public trading mechanism in which goods of merchandise are sold by a seller or an autioneer to a buyer who offers the highest bid (Bajari *et al.*, 2009).

Consequently, planning the delivery of a service involves three main activities.

- Discovery of resources
- Negotiation for service delivery
- Creating of a service plan

A client or a service requestor first finds the resources or service providers that can supply the services required by the client. Then the client negotiates with the resources for a specific episode of service delivery that is based on the needs of the client and the constraints of the resources. Finally, when there is agreement between the negotiation parties, the service plan is created .

The role of information within the context of planning, therefore, is different from an information gathering context where the information is used to describe the actual service, and so acts as a means of accessing the actual service. Here, the role of negotiation is to agree on service provision, rather than to perform access control as occurs in the context of information gathering.

When compared with 'manual' service planning, a planning service system or tool, developed as service-based software, can provide advantages for the planner from several perspectives, and especially.

- When dynamic ‘on the fly’ choice of services is needed

- To permit faster emergence of negotiation agreements

The planning system enables the planner to produce and manage the plan by dynamically finding and negotiating with a number of various providers in the emergent global market. So the planner can get the most suitable services from the providers. Moreover, the planning system can set up terms for automatic negotiation where services are selected dynamically.

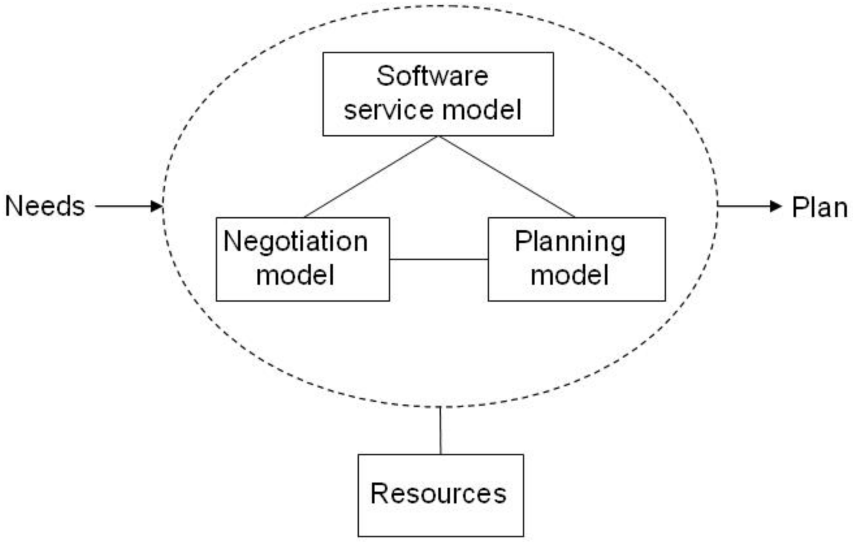


Figure 1.3: The conceptual framework of planning service system

Figure 1.3 shows the conceptual framework of the planning service system. The conceptual framework merges planning, negotiation and software service-based models in order to provide planning services to produce a plan for the client. The plan identifies the most appropriate resources for supplying the set of services that are requested by the client.

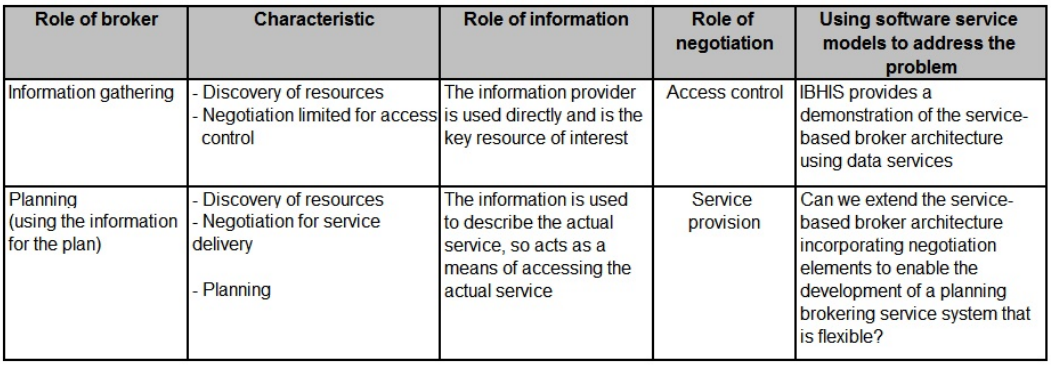
The three models of the conceptual framework are employed for different objectives as follows.

- The planning model is based on the particular planning service context, such as health and social care planning.
- The software service model provides the dynamic service context that the system employs to find the candidate resources and invoke their services.
- The negotiation model allows the plan to be adapted to meet the specific needs of the client.

The planning model is the central structure of the planning system for organising resources to meet the needs of the client. The software service-based model enables the planning system to find and use the resources as they become available to provide their services to the client. The negotiation model enables adaptation of the details of the plan, based on the interest of the client, by interacting with the providers. Consequently, these three models can be used for developing a flexible planning service system that can deal with the dynamic needs of the client and the constraints of the providers.

1.1.3 Research Question

Table 1.1 compares the role of the broker in these two contexts of information gathering and planning. They are different in terms of problem; characteristic; the role of information and negotiation; and the method of using software service models to address the problems.



Role of broker	Characteristic	Role of information	Role of negotiation	Using software service models to address the problem
Information gathering	- Discovery of resources - Negotiation limited for access control	The information provider is used directly and is the key resource of interest	Access control	IBHIS provides a demonstration of the service-based broker architecture using data services
Planning (using the information for the plan)	- Discovery of resources - Negotiation for service delivery - Planning	The information is used to describe the actual service, so acts as a means of accessing the actual service	Service provision	Can we extend the service-based broker architecture incorporating negotiation elements to enable the development of a planning brokering service system that is flexible?

Table 1.1: Research context

Therefore, to acquire a planning system, the service-based broker architecture needs to include a proper negotiation model alongside the planning model, so that the planning system developed is flexible and adaptable. Consequently, the development of a planning system based on the service-based broker architecture leads to the research question.

> "*Can we extend the service-based broker architecture incorporating negotiation elements to enable the development of a planning brokering service system that is flexible?*"

This question essentially asks whether a software service-based model can be used to provide a flexible and agile software providing services, particularly information gathering, to meet the needs of emergent open markets. The service-based model can also include the concept of the broker to act on behalf of the end user in order to deal with a number of agencies, based on the interest of the end user. However, the broker and the agencies may need some degree of interaction, especially service negotiation, to accomplish a specific task for the end user. Hence, the planning broker provides planning services by finding appropriate agencies, negotiating with them, and creating a service plan for the end user.

In addition, the provision of service to a consumer may be supplied by several service providers through a service supply chain (Bennett et al., 2000). From the lowest layer of the service supply chain, suppliers supply their basic or primitive services through a hierarchy of service providers or a consumer-supplier (retail) market. The consumer, at the upper end of the service supply chain, should receive the service that meets the consumer's needs. Therefore, the information broker can also form a primitive service in a hierarchy of service providers by providing its information to the other service brokers, through a consumer-supplier (retail) market. A planning service broker in the higher layers of the hierarchy of service providers does not have to deal with the original data sources directly and can focus on their roles and let the information broker deal with the distributed, heterogeneous and autonomous data sources.

We therefore propose to investigate how a service-based broker architecture, that includes the concept of the broker, the service supply chain and the negotiation model, can be extended for the development of a flexible and adaptable planning system, providing the planning services for the end user.

1.2 Research Method

The research method adopted for this project is based on "Concept Implementation" as described in Glass *et al.* (2004). The objective is to create a 'proof of concept' system to demonstrate the feasibility of a research. This project aims to develop and investigate a service-oriented broker system, called CAPTAIN (Care Planning Through Auction-based Information Negotiation), by extending the concepts and role of the broker used in IBHIS. CAPTAIN also extends the service negotiation function employed in IBHIS in order to demonstrate a full range of service-based software characteristics: description, discovery, negotiation, delivery and composition. A service-oriented broker architecture for CAPTAIN has been developed based on a case study by identifying the specific issues and problems of the case study within healthcare context. The CAPTAIN service-oriented broker architecture includes a negotiation model that has been developed for the service negotiation. A CAPTAIN prototype has been built and implemented, based on the service-based broker architecture. The prototype is used as a “proof of concept” system to demonstrate the feasibility of the proposed model. The prototype has been demonstrated through application of a use case and its scenarios, focusing on the three main features of the service-oriented broker application: function, data access and negotiation. The demonstration application is used to evaluate the potential output and outcome of the extended service-oriented broker architecture.

Consequently, this research uses the idea of an integrated care plan, as deployed within the healthcare sector, to provide a case study. A clinician, who acts on behalf of a patient, determines the form of a plan for a patient and operates the plan that will satisfy the patient’s needs. The clinician then negotiates with several health and social care providers in order to agree on the terms and conditions relating to supplying the services need for plan.

To achieve the objective of this research, the software service-oriented broker architecture for CAPTAIN employs two forms of service broker.

- A care planning broker or *P-Broker* providing a care planning service for the integrated care plan needed by a client
- An information broker or *I-Broker* providing an information service for the P-Broker

The P-Broker negotiates with various health and social care service providers based on the needs of the client. During service negotiation, the I-Broker acts as an intermediary by organising exchange of the messages between the P-Broker and the service providers. Data access in CAPTAIN takes the form of an active role (read-write), as when there is an agreement between negotiation participants, the content in the data resources will need to be updated to represent the new state.

CAPTAIN employs a negotiation model through which the P-Broker negotiates with the service providers to produce the integrated care plan that meet the client's needs. This negotiation model uses an auction model, and is based on the conflict of interest between the client and the service providers. It was developed based on the ideas of the traditional negotiation model together with several more specific negotiation models, especially Web services (Hung *et al.*, 2004), service-oriented environments (Elfatatry and Layzell, 2004) and E-business (Kim and Segev, 2003).

The traditional negotiation model consists of three fundamental negotiation elements: object, protocol and decision model (Jennings *et al.*, 2001). In addition, a specific negotiation model may then be needed to deal with a particular negotiation situation because different negotiation situations have different characteristics (Bichler *et al.*, 2003). Therefore, the negotiation model for CAPTAIN consists of the following four main negotiation elements.

1. *Negotiation object* contains the information needed for negotiation between the P-Broker and the DASs in order to produce the integrated care plan for the client.

2. *Negotiation protocol* specifies the rules and the states of negotiation interaction that all system components of CAPTAIN have to follow during service negotiation process.

3. *Negotiation process* is a dynamic end-to-end process starting from the client request and ending up with a resolution. There are three main stages of negotiation process: pre-negotiation, service negotiation, and post-negotiation.

4. *Decision model* comprises of the rules employed by the P-Broker and the DASs for decision making during their negotiation.

These negotiation elements are incorporated into CAPTAIN so that the CAPTAIN system can provide a full range of service characteristics: description, discovery, negotiation, delivery and composition.

The implementation of the service-oriented application is based on Java 2 Enterprise Edition (J2EE) Web Services technologies, and runs within Eclipse Java EE IDE for Web Developers environment working on the Windows platform. The application supports current Web services standards, for instance SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration) and XML (Extensible Markup Language).

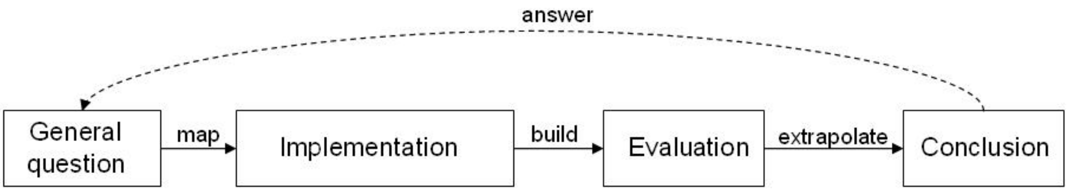


Figure 1.4: Research framework

Figure 1.4 shows the overall research framework for this thesis. It involves four main stages described as follows.

1. *General question*

   The general question from Section 1.1.3 addresses the context of the software service-based model and the planning services. It is initially motivated from experience with IBHIS, a service-based broker software. The general question is mapped to the implementation part that uses the integrated care plan as a research case study for the service-based broker model of the planning services and service negotiation.

2. *Implementation*

   A service-based broker architecture for CAPTAIN has been designed and developed by extending the concepts and role of the broker used in IBHIS. The prototype of CAPTAIN has been developed and implemented in order to demonstrate a 'proof of concept' for the use of negotiation. The development of the prototype consists of five main parts as follows.

- Develop the system architecture of CAPTAIN
- Develop a negotiation model for CAPTAIN
- Define the key issues and problems of the integrated care plan
- Incorporate negotiation elements into CAPTAIN
- Develop the prototype of CAPTAIN

3. *Evaluation*

   A comprehensive use case and associated scenarios are built based on the integrated care plan. They are used to test and evaluate the CAPTAIN prototype.

4. *Conclusion*

   The results of evaluation are extrapolated to derive conclusions about how well the use of services and negotiation are able to answer the research question.

The result of this project is a service-oriented planning broker application. It is a planning system that is flexible, adaptable and platform independent. It deals with the needs of the client and the constraints of the providers within planning service-oriented context. The output of the planning system is an integrated care plan that meet the client's needs.

This project provides two major outcomes. Firstly, it demonstrates the scalability of the software service-oriented broker architecture. The new planning broker service can potentially and flexibly extend the concept of the service-based broker architecture, through the service supply chain within a software services open market. As a result, the extended service-based architecture can be used as the basis for diverse kinds of service-oriented broker software.

Secondly, the flexibility of service interaction between the service brokers is the other key outcome of this project. The negotiation model can be employed to accommodate the service broker in order to compose and tailor with the other broker services to deal with their conflict of interests and/or to obtain the best offered services from a group of the candidate service brokers.

Therefore, the concept of the service-oriented broker architecture incorporating elements of negotiation can be extended to enable a scalable and flexible distributed information management with interaction.

1.3 The Criteria for Success

In this thesis, the criteria for success have been defined as follows.

1. The functions of the CAPTAIN broker

   The CAPTAIN broker provides a care planning service that performs as the central element of the CAPTAIN system. The client should be able to use and operate the system, through the CAPTAIN broker, to create, modify and manage the integrated care plans that should meet the needs of a patient.

2. The role of data access (read-write)

   The planning broker provides a care planning service by negotiating with the providers to create the integrated care plan to the patient. The negotiation of both negotiation participants is based on content. The content in the data sources should be updated if the planning broker and the providers have an agreement on terms and conditions of the providers' service provision.

The information broker provides information services to the planning broker. Therefore, to meet a request from the planning broker, the information broker needs data access (read-write) to the providers' data sources in order to update the status of the service negotiation.

3. The negotiation model

   The negotiation model is the key part of the service negotiation between the CAPTAIN broker and the service providers. During service negotiation, all negotiation elements should perform properly to achieve the objective of the negotiation participants. The result of the negotiation is the integrated care plan required and should be satisfied by the patient.

4. The system behaviour of the CAPTAIN prototype

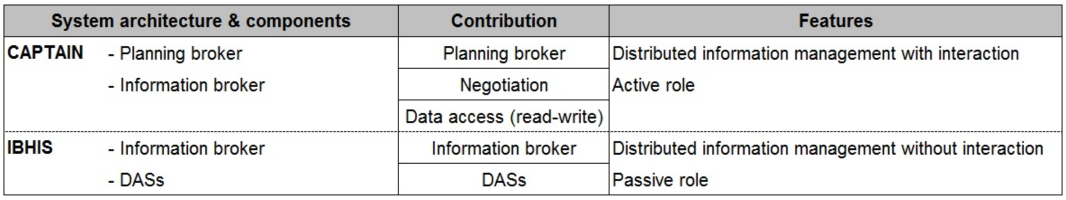
   The CAPTAIN system is a distributed system consisting of different system components which are spatially separated. The system behaviour of CAPTAIN is observed in terms of the main system components to ensure that they process their tasks, and then, send and receive messages among them as in the event orders specified or regulated by CAPTAIN. An event logger should be used to record the timestamp of a sequence of the events, occurring when there are the execution of a subprogram or interactions among system components of CAPTAIN.

5. Use case and scenarios

   A set of scenarios based on the use case is used to evaluate the CAPTAIN prototype based on its three key features: the functions of the broker, the role of data access (read-write), and the negotiation model. The evaluation result should also reflect the potential of the extended service-oriented broker architecture.

1.4 Contributions

Table 1.2 shows the service architectures, main contribution and features of CAPTAIN and IBHIS repectively. The IBHIS service architecture consists of the information broker and DASs. IBHIS involves distributed information management without interaction in which the DASs perform a passive role in providing data services to the information broker. The content of the DASs' data sources is accessed by the information broker based on read-only protocols.



System architecture & components		Contribution	Features
CAPTAIN	- Planning broker	Planning broker	Distributed information management with interaction
	- Information broker	Negotiation	Active role
		Data access (read-write)	
IBHIS	- Information broker	Information broker	Distributed information management without interaction
	- DASs	DASs	Passive role

Table 1.2: The contribution of CAPTAIN

The contributions of CAPTAIN are the extension of the concepts and role of the broker, the service negotiation and data access model used in IBHIS. The CAPTAIN service architecture encompasses a care planning broker to provide planning services to an end user. CAPTAIN deals with distributed information management with interaction in which the DASs perform an active role to negotiate with the planning broker to accomplish the objective of the end user. Service negotiation needs data access with the 'read-write' property to the DASs through the information broker in order to update the status of service negotiation.

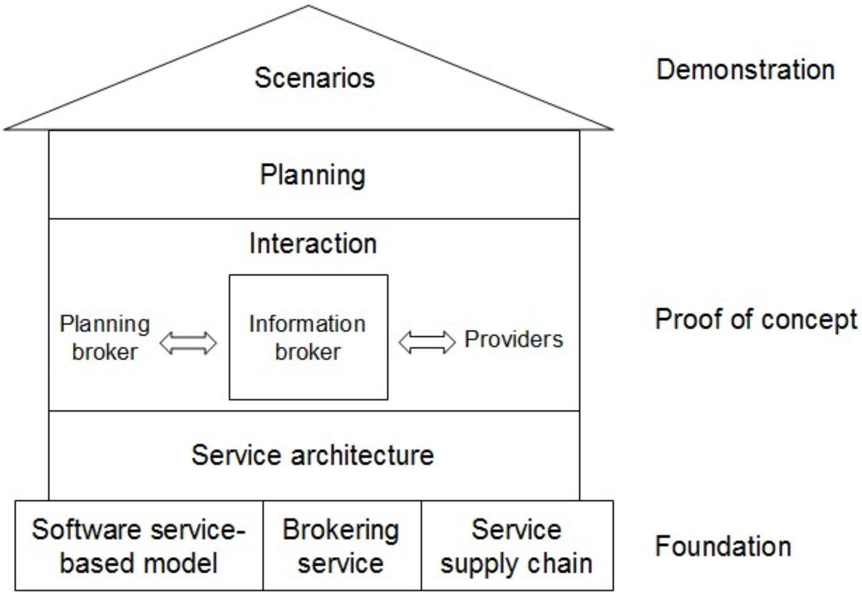


Figure 1.5: The framework of the concept implementation

Figure 1.5 presents the framework of the concept implementation to provide a 'proof of concept' of CAPTAIN. The key contribution of the concept implemetation is described as follows.

1. Software service-oriented broker architecture

   The software service-oriented broker architecture for CAPTAIN is developed based on three major foundations: software service-based models, brokering services and a service supply chain. The architecture includes planning, negotiation and service-based software models. This architecture enables the development of a flexible and adaptable service-oriented broker system providing care planning services to the end user.

2. Planning model

   A planning model is developed based on the use case of an integrated care plan within health and social care context. The objective is to enable the development of the planning service system that produces and manages the integrated care plan for the end user.

3. Negotiation model

   A negotiation model has been developed based on the integrated care plan and an auction model. The negotiation model provides the interaction between the planning broker and the providers via the information broker in order to deal with their conflicts of interests. The aim of the negotiation interaction is focused on the end user's needs or preferences. The elements of negotiation for CAPTAIN have been chosen based on the conventional negotiation model and several automated negotiation models from various aspects of negotiation situations, such as service-oriented environments and Web services. The negotiation model enables the adaptable planning service application to produce the integrated care plan, based on the interest of the end user.

4. Service-oriented broker application

   The development of the service-oriented broker application is based on the software service-oriented broker architecture that incorporates the planning, negotiation and service-based broker models. The CAPTAIN broker provides a flexible and adaptable planning services producing the integrated care plan according to the needs of the client. The application provides a proof of the extended broker concept.

5. Use case and scenarios

   A set of scenarios based on the use case has been developed based on an integrated care plan within the context of health and social care. The scenarios are used to demonstrate, evaluate and review the service-oriented broker application of CAPTAIN as the "proof of concept". The use case developed focuses on the three main features of the service-oriented broker application: the functions of the broker, the role of data access (read-write), and the negotiation model.

1.5 Overview of the Thesis

This thesis is organized into nine chapters.

Chapter 1 discusses the problem to be solved. It also describes the motivation and aim of this research and defines the method used to investigate and to develop a service-based broker concept and its application. Then it sets out the criteria for success as well as its contribution and outlines the structure of the thesis.

Chapter 2 examines the background of service-based software. It describes the service concept in software, following with service-based software. Two main software service models are then introduced: Service-Oriented Architecture (SOA) and Software as a Service (SaaS). Web services, as a form of service software technology, are described, as well as an explanation of the relationship between SaaS and Web services.

Chapter 3 describes the architecture and operation of the IBHIS broker, which is a ‘proof of concept’ of SaaS. This chapter also decribes how the IBHIS broker was implemented by using Web services.

Chapter 4 examines the studies of negotiation from various perspectives. It describes the principles of negotiation, as well as the types of negotiation studied and applied in different aspects. Then the role of automated negotiation is discussed and we introduce some negotiation models that are useful for providing a negotiation model for CAPTAIN.

Chapter 5 determines the nature of the integrated care plan that forms the basis of the research case study. This chapter presents the conceptual framework and the operation of a care planning system used for producing the integrated care plan for a client. An example care plan is presented and described by means of a Use Case and its scenarios.

Chapter 6 describes the implementation of the broker concept of CAPTAIN. It presents the architecture of CAPTAIN, including its negotiation model. Negotiation elements of the negotiation model are described. This chapter describes the implementation of the CAPTAIN application, starting from the original prototype to the present CAPTAIN application. It describes the development and deployments tools used for developing and implementing the prototype CAPTAIN application.

Chapter 7 evaluates and reviews the results from using the 'proof of concept' version of CAPTAIN. It describes a comprehensive use case and associated scenarios based on the integrated care plan, that are used to evaluate the CAPTAIN system. The results of the evaluation are also described to show that CAPTAIN system can perform the main service functions of creating, modifying and deleting an integrated care plan that meet the purpose of this research. During and after negotiation, the data sources are accessed to update the content based on the agreements between negotiation participants. In addition, the negotiation model enables adaptation of the details of the plan, based on the interest of the client, by interacting with the providers.

Chapter 8 discusses and interprets the evaluation of CAPTAIN. It also reviews several issues such as its limitations and discusses how well the research question is answered.

Chapter 9 presents the conclusion of this thesis, summarizes its original work and contribution, and then identifies some possible directions for further work.

Chapter 2

Service-based Software

This chapter examines the background of service-based software. It also describes how the service concept is realised in software. Then it describes and compares the specific software service models involved in this thesis. Finally it describes technologies that are used to realise software services.

2.1 The Service Concept in Software

A service is described as "a software component of distinctive functional meaning that typically encapsulates a high-level business concept" (Krafzig et al., 2004). A service's aim is to take on particular functions to produce value to the business (Domingue et al., 2009). Its specification can be classified in terms of a number of characteristics, such as service contract that meets a consumer's different choices, and such as security addressing authorisation for accessing and using a particular service. Hence, the service is a provident domain of control containing a group of tasks to accomplish any objectives related to the service provision.

A service has the following general characteristics (Domingue *et al.*, 2009).

- *Intangibility*: Services are intangible in that delivery is different from that used for goods which are physical.

- *Perishability*: Both perishable services and goods have similar characteristics that they cannot be used after expired date and kept for later use.

- *Customer contacts*: A customer, who contacts a provider for services more often, requires the higher rapid demand for a service.

- *Simultaneity*: A service consumer is not just a person who expresses a demand for products and services any more. Instead the consumer has become a part of the production process, as a co-creator of a service.

- *Heterogeneity*: The heterogeneity of customers' service demands reflects the frequency of interaction between the service provider and the consumer. Some consumers require a simple service while the other consumers needs highly elaborated services.

- *Demand fluctuation over time*: The level of capacity management required depends on the variation in demand. Therefore the level of service production is in response to its consumption.

- *Customization*: Within the competitive environment, customized services based on the consumer's needs are a value-added feature that attracts a consumer to purchase the service.

- *Complexity*: Simultaneity of production and consumption of services leads to higher interaction among system components, involving people, processes and shared information, leading to greatly complicated service systems.

The provision of service to a consumer may be supplied by several service providers through a service supply chain (Bennett et al., 2000). From the lowest layer of the service supply chain, suppliers supply their basic services through a consumer-supplier (retail) market or a hierarchy of service providers. The consumer, at the upper end of the service supply chain, should receive the service that meets their needs. There may be service contracts or agreements between the consumer and the service supplier(s) that specify the terms and conditions for buying and selling the services. These may involve binding legal agreements. So the supplier can guarantee that the consumer will receive services that meet the quality of service levels defined in the contracts or agreements.

In terms of delivering software services, services are classified as being of two main kinds (Kohlborn, Korthaus and Rosemann, 2009).

1. *Atomic services* are independent from other services. Atomic services supply their functionality to the other services or the client. There are three types of atomic services.

   (a) Utility services provide application functionality that is relevant to data processing within legacy system environments.

   (b) Entity services present business centric services. They include one or more services needed to generate a business interface for an end-user.

   (c) A task service is one that is created in order to accomplish a firm's ad hoc needs.

2. *Composite services* consist of the following two types.

   (a) Logic-aggregation services manage and provide services for the end-user by composing and processing a group of atomic services based on process logic and related business rules.

   (b) Data-aggregation services encompass entity services to provide their functionalities as a package to integrate information, exposed across a number of heterogeneous systems with different data models embedded in them.

Consequently, end-user services can be composed out of smaller ones (and so on recursively) and in a dynamic context, these are procured (and possibly paid for) on demand (Bennett et al., 2001). The end-user can create, compose and assemble services by bringing together a number of suppliers to meet the end-user's requirements at a specific point in time.

2.2 Service-Based Software

The role of a service in the new era of modern business has obviously been changed from the forms of service provided by traditional businesses. This has mainly resulted from the rapid growth of the internet technology. The internet has been increasingly used as a *disruptive platform* for new various kinds of business model (Domingue *et al.*, 2009). However, the conventional development of software has focused on supply-side issues, driven by developers of technology rather than end users (Budgen *et al.*, 2004). Hence, software developed and delivered as a product is not adaptable to the new era of rapidly changing business needs. To deal with new demands of the consumers, software can be delivered as a service, focusing on the quickly changing needs of the consumer. As a result, the software industry as established by software vendors has begun to move from software products to more profitable software services (IBM-SaaS, 2010).

A software service is the delivery of software functionality in a flexible and adaptable approach by using autonomous and distributed computational elements, usually across a appropriate high-bandwidth network (Budgen et al., 2004; Bennett et al., 2000). A software service exhibits a set of reusable application functionalities that can be composed in regard to business requirements (Kohlborn, Korthaus, Chan and Rosemann, 2009). The resulting benefit is the higher level of flexible and agile software to meet the needs of rapid changing business. A software service can be essentially characterised as follows (Budgen *et al.*, 2007).

- Being used rather than owned
- Being independent of programming language construction or platform
- Being stateless and not preserving user-related knowledge across episodes of use
- Providing a contractual interface

Services are regarded as a means of providing the “service-oriented” approach that is independent from particular software programming languages or operating systems (Papazoglou *et al.*, 2007). The “service-oriented” approach allows software to be composed by discovering and invoking a set of services through a network of services. Use of service orientation enables business services or operations to be exposed and offered to other service software or business partners in order to facilitate on-demand collaboration opportunities (Kohlborn, Korthaus, Chan and Rosemann, 2009). Hence, “service-oriented approaches are used for developing software applications and software-as-a-service that can be sourced as virtual hardware resources, including on-demand and utility computing” (Demirkan *et al.*, 2008).

Consequently, a service-oriented model of software can be used to develop software that provides a service that is consumed on demand and may be discarded later by the end users (Budgen *et al.*, 2004; Elfatatry & Layzell, 2004; Demirkan *et al.*, 2008). This new approach to software development has employed a demand-led strategy that focuses more on the needs of the end users or service consumers (Kohlborn *et al.*, 2009; Budgen *et al.*, 2004). Software service-based models have been used as the means to develop software around a demand-centric aspect that leads to software delivered as a service (Bennett *et al.*, 2000; Kohlborn *et al.*, 2009). “The most significant benefit of the service model is the ability to loosely couple dynamic services, finding and binding the services as needed” (Budgen *et al.*, 2007). An example of the evolving nature of software service-oriented models is the concept of a service-oriented architecture (SOA) (Laplante *et al.*, 2008; Papazoglou *et al.*, 2007). An SOA is used to design and develop an agile service-oriented system, providing services that are dynamically and loosely coupled and that are consumed by other distributed applications or services.

In the service-based model of software, there are two main following issues that need to be considered: the nature of the service supply; and the anatomy and structuring of services (Bennett et al., 2000).

1. The nature of the service supply

   The nature of the service supply is the means by which services are provided through a service supply chain from primitive services to consumers. The primitive services, at the bottom of the chain, provide essential system functionalities. They are supplied by a hierarchy of service providers: supplier-supplier (wholesale) and consumer-supplier market (retail). At the top of the chain, consumers with needs are supplied and satisfied by the provision of software services. In an entirely adjustable service supply chain, consumers would not need to renegotiate for service provision before the time of need because of ultra-late binding between the business problem and solution. Service components are bound instantly, as when they are needed, and then that binding is discarded. Non-functional attributes of the software are required by ultra-late binding to support automatic negotiation and resolution.

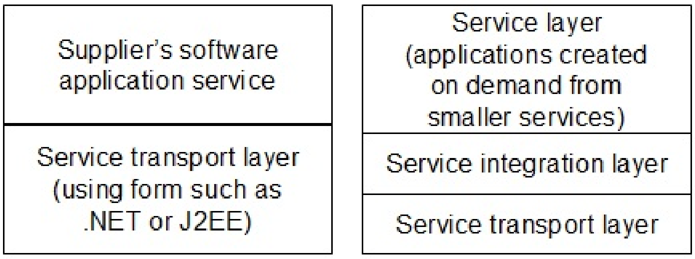
2. The anatomy and structuring of services

   For the anatomy and structuring of services, the structure of software must be finer-grained and operate transparently in order to be flexible software required by a service-based environment. Consequently, non-technical factors, such as a service agreement, can apply to the entire software product. Different terms and conditions of its use may relate to specific parts of a software product. Hence, a service can be described as being a highly interrelated software element requiring combination which is wrapped by a service-level agreement.

2.3 Software as a Service

SaaS is a service-based software model. It has been observed that “clients are increasing their adoption of SaaS, driven by ease of deployment, flexibility, scalability, and predictable pricing models” (IBM-SaaS, 2010). SaaS is based on a demand-led concept in which desired services are assembled and used, as and when needed, to deal with a specific requirement (Budgen et al., 2004). It focuses on detaching the possession and ownership of software from its use by configuring and joining a set of services at the time of delivery. “Using SaaS, a vendor can deliver a software system as a service” (Laplante et al., 2008). The advantage of delivering software services with this approach is in reducing the limitation of software use, deployment and evolution, for instance cost and time.

In the SaaS model, “services are composed out of smaller ones (and so on recursively), procured and paid for on demand” (Bennett et al., 2001; Budgen et al., 2007). Composing such a service involves managing supplier-consumer relationships. Users can create, compose and assemble a service by bringing together services from a number of suppliers to meet needs at a specific point in time.


Supplier’s software application service
Service transport layer (using form such as .NET or J2EE)
Service layer (applications created on demand from smaller services)
Service integration layer
Service transport layer

(a) Supply-led (b) Demand-led

Figure 2.1: Layered service model taken from (Turner et al., 2003)

Figure 2.1(a) and 2.1(b) presents the service supply-led and demand-led model respectively (Turner et al., 2003; Budgen et al., 2004). The supply-led service model consists of two layers. The service transport layer employs technologies such as Microsoft's .NET or Sun's J2EE platform together with XML-based envelopes and message formats, for example Web services Description Language (WSDL). A set of composed services from this layer delivers a pre-defined set of applications at the top layer.

However, the applications provided by the supply-led model are not flexible because they only provide a pre-defined range of services from the service providers. The techniques, processes and methods associated with software development based on the supply-led model have been conventionally focused on supply-side matters (Bennett et al., 2000; Elfatatry and Layzell, 2004). This can lead to the production of software that is oriented towards the perceptions of software developers instead of the needs of end-users. However software products or applications developed in this way are not necessarily suitable for an open marketplace or emergent organisations that are in a state of constant process change (Cusumano, 2008; Bennett et al., 2000). Hence the argument that software should deliver a service in order to accomplish the aims of functionality, flexibility and time to market or to organisations (Turner et al., 2003). This new approach to software development is therefore focussed on a demand-centric aspect, so that software can be delivered as a service within the framework of a global marketplace (Kohlborn, Korthaus and Rosemann, 2009; Bennett et al., 2000, 2001).

The demand-led model is extended from the supply-led model. The demand-led model is flexible, extendable and adaptable because it includes a service integration layer over the service transport layer (Budgen et al., 2004). Applications can be built from a set of atomic services and bound dynamically as needed. The service integration layer comprises of five major service functions (Budgen et al., 2007).

1. *Service description* provides several elements of the elementary information about the services provided. It includes descriptions of such characteristics as functionality, non-functional aspects, constraints, and interfaces. Additionally, it expresses the parameters used to negotiate by the client and the provider.

2. *Service discovery* provides the means by which appropriate services will be located by the client.

3. *Service composition* is the means by which a service is composed from a set of lower level services.

4. *Service negotiation* is a process in which the client and service providers interact to reach an agreement on the terms and conditions for the supply of a service.

5. *Service delivery* is the process of actually delivering an agreed service. It consists of three sequential steps: invocation, provision and suspension.

2.4 SaaS and SOA

A service-oriented architecture (SOA) embodies a software service-based model and the concept has been getting a lot of attention as the means of developing service-based software. However, the concepts of SOA and SaaS may make people confused about the difference between SOA and SaaS. This section will describe the concept of SOA and the differences between the two concepts as follows.

2.4.1 Service-Oriented Architecture (SOA)

A service-oriented architecture (SOA) is a software service-based model in which is “a business-driven IT architectural approach that supports agile business innovation and optimization" (IBM-SOA, 2007) by "integrating business as linked, repeatable business tasks, or services" (IBM-SOA, 2010). The SOA quickly and easily assembles, related business functions into the particular application required by the business at a given point of time, and is important for meeting the on demand business needs for flexibility and integration. An SOA provides "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" (Kohlborn, Korthaus, Chan and Rosemann, 2009). It is based on a form of a distributed systems architecture (W3C, 2010; Booth et al., 2004), where a distributed system is composed of various and discrete software systems or agents that must cooperate to perform some tasks. These software systems may all not operate in the same processing environment. Therefore, they have to communicate through a network, based on hardware/software protocol stacks.

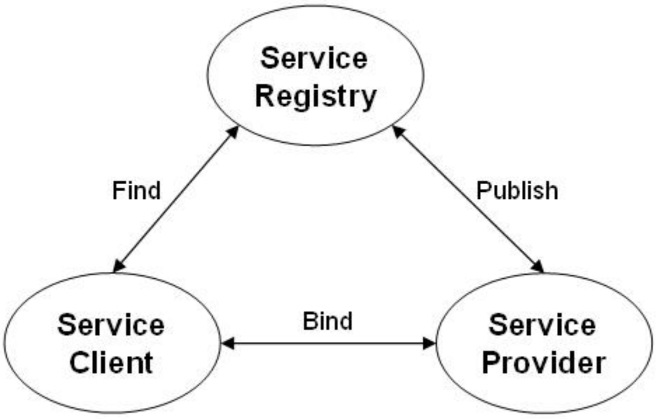


Figure 2.2: Service Oriented Architecture taken from (Erl, 2007)

Figure 2.2 shows the relationship of three types of SOA participants: the service provider, the service registry, and the service client. The interactions between these participants address the publishing, finding and binding operations (Erl, 2007).

1. The service provider specifies a service description and publishes it to a service registry.

2. The service client or requestor can retrieve the service description from the

service registry to find the service provider.

3. It uses the service description to bind with the service provider and then invokes the service through the service implementation.

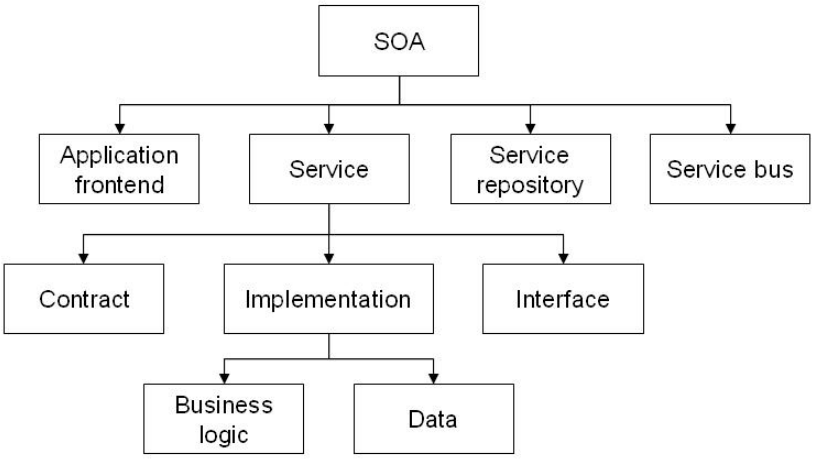


Figure 2.3: The key abstraction of an SOA taken from (Krafzig et al., 2004)

Figure 2.3 illustrates the four main abstractions of an SOA that are described as follows (Krafzig et al., 2004).

1. The application frontend is the proprietor of the business process, providing business services to the client.

2. A service supplies business functionalities that are employed and used by the application frontends and other services and consists of the three following main components.

   (a) A service contract that identifies the functionality and forms of use, together with terms and conditions between the client or service consumer and provider to use the service.

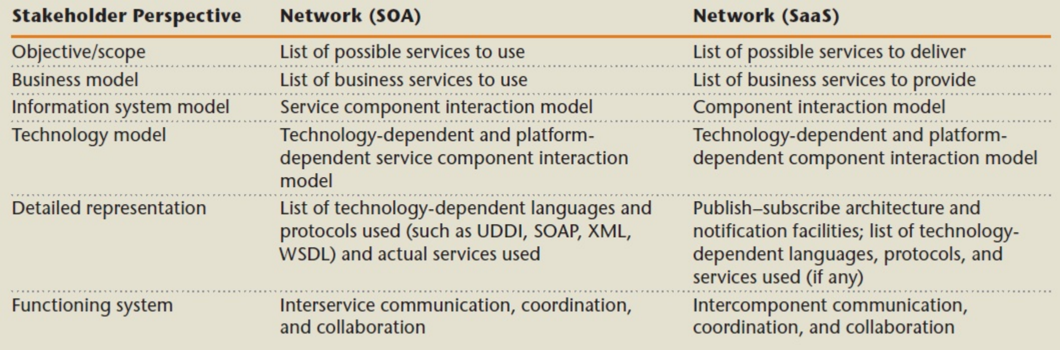
   (b) A service implementation provides business logic and data according to the needs of the application frontends and other services.

(c) A service interface physically provides the functionality, through which the client of the application frontends may interact with the services.

3. A service repository retains the service contracts of the particular services of the SOA.

4. A service bus accomodates the interconnection between the application frontends and services.

2.4.2 Comparison Between SOA and SaaS



Stakeholder Perspective	Network (SOA)	Network (SaaS)
Objective/scope	List of possible services to use	List of possible services to deliver
Business model	List of business services to use	List of business services to provide
Information system model	Service component interaction model	Component interaction model
Technology model	Technology-dependent and platform-dependent service component interaction model	Technology-dependent and platform-dependent component interaction model
Detailed representation	List of technology-dependent languages and protocols used (such as UDDI, SOAP, XML, WSDL) and actual services used	Publish–subscribe architecture and notification facilities; list of technology-dependent languages, protocols, and services used (if any)
Functioning system	Interservice communication, coordination, and collaboration	Intercomponent communication, coordination, and collaboration

Table 2.1: Focused Zachman model for comparing SOA and SaaS taken from (Laplante et al., 2008)

Laplante et al. (2008) uses the Zachman architectural model to distinguish the SOA and SaaS approaches to building software, as shown in Table 2.1. “Because the Zachman model is so intuitive, the approach” taking “to describe the differences between SaaS and SOA works well even with non-IT professional” (Laplante et al., 2008). The comparison of SOA and SaaS focuses on the network dimension in which service components of SOA and SaaS architecture are connected via a network. The comparisons are described from the following perspectives.

1. Objective/scope

   The SOA is a group of a software services that can be potentially used . While the group of services for SaaS is meant to be delivered.

2. Owner

   From the SOA point of view, a set of services is be found and used. However SaaS describes the business services to be provided. The advantage of using existing business services greatly reduces the cost of software design and development.

3. Designer

   The SOA is viewed as an architectural model used to describe how constituent service components interact with each other. On the other hand, SaaS depicts the forms of interaction that occurs among consituent components, which are not typically services.

4. Builder

   Both SOA and SaaS have to specify a technology, for instance Web services, is suited to the interaction patterns identified in the information system model.

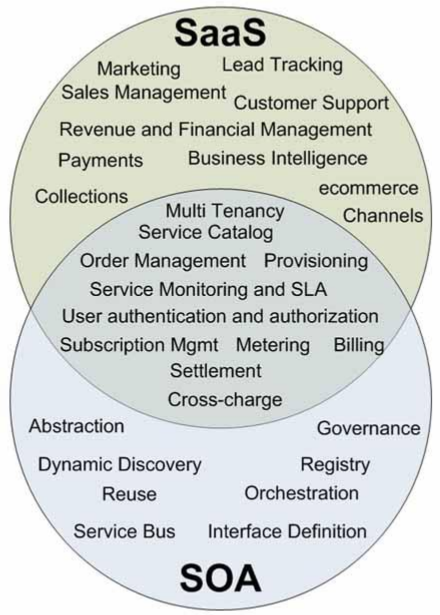


Figure 2.4: The overlapping concepts of SaaS and SOA taken from (Singla, 2009)

In addition, Singla (2009) presents the overlapping concepts of SOA and SaaS as shown in Figure 2.4. The similarity and difference of both SOA and SaaS are described from the aspects of their exposure, deployment and the key elements of platform.

1. Exposure

   SaaS is regarded as the provision of applications over the network based on a subscription basis with the pay-as-you-go model. SOA is primarily used to bring out functionality from distributed systems in the format of stateless functions.

2. Deployment

   SOA deployment are commonly carried out within IT environments and the services are provided to end users via the other applications. In a different manner, SaaS deployments create business processes that are aimed directly at the end users.

3. Platform

   A SaaS platform consists of two following key elements.

   (a) Core elements for supporting main business functions, such as ordering and provisioning; user authentication and authorization; and billing.

   (b) Supporting elements for the common business functions, such as marketing; lead tracking; and revenue and financial management.

   The general SOA platform comprises of two main elements consisting of service providers and consumers from diverse business organisations. The providers publish their services through the SOA platform, and these are then used by the consumers. Moreover, from the technical perspective of the SOA platform, there are a number of other elements, for instance communication

> protocol (Simple Object Access Protocol - SOAP), service interface definitions (Web Services Description Language - WSDL) and service discovery (Universal Description, Discovery, and Integration - UDDI).
>
> Furthermore, a typical large enterprise consists of the service providers and consumers, that could be applications or systems owned by various departments or organisations. Therefore a proper service management process is needed to deal with service provision and consumption on the service providers' host that takes into account of cost of creating and/or using services. Consequently, service management with cost leads to the need for a number of services, such as service catalogue management, authentication and usage metering. These services deployed by the SOA are also become the need of the core elements of a SaaS platform.

"Brieftly stated, the difference between SaaS and SOA is that the former is a software-delivery model whereas the latter is a software-construction model" (Laplante et al., 2008). "It is probably quite intuitive that most complex architectures including SaaS architecture will benefit from SOA capabilities, but an SOA platform needing SaaS capability is not that intuitive" (Singla, 2009). Consequently, the SOA adopts a service component to construct new service systems that are published and delivered as new services. Then SaaS delivers the software service system as a service to a client (Laplante et al., 2008).

2.5 Web Services

"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network" (W3C, 2010). Web services provide technologies for "programmable Web applications with standard interface descriptions that provide universal accessibility through standard communication protocols" (Kohlborn, Korthaus, Chan and Rosemann, 2009). Web services technologies have been used as the technological implementation of a service-based model of software to provide software services.

Web services provide an approach for building and deploying distributed computations by enhancing the analogous productivity of developers, administrators, and end-users (Singh and Huhns, 2005) Their goals are to supply a range of functionalities on behalf of its owner, a person or organization (W3C, 2010; Booth et al., 2004). A web service can be employed by an agent which is a computational resource. It corresponds to the notion of a software agent. The agent carries out tasks on behalf of individual or organization by exchanging messages among them over the network. There are two main types of the agent's entity: a provider entity and a requester entity. Both entities exchange messages by using a requester agent and provider agent of the requester entity and of the provider entity respectively.

- A provider entity supplies an agent to employ a specific service.
- A requester entity demands and uses a provider entity's service.

From a technical point of view, Web services are a standardized approach to integrating Web-based applications (W3C, 2010). Web services employ a number of open standards, especially:

1. Extensible Markup Language (XML)

   XML is a common language for structuring messages exchanged between service requestors and service providers.

2. Simple Object Access Protocol (SOAP)

   SOAP provides the common communication protocol used for transferring the messages.

3. Web Services Description Language (WSDL)

WSDL describes the available services retained in a service registry. It specifies the message formats, datatypes, transport protocols, and transport serialization formats used between the service requester agent and the service provider agent. Moreover it defines one or more network locations where a provider agent can be invoked.

4. Universal Description, Discovery, and Integration (UDDI)

   UDDI specifies a registry of services which found by the service requestors.

Therefore Web services can be used to develop and deploy web applications producing software services for the service providers and consumers. Consequently, the service providers can describe and publish the service descriptions so that the service consumers can find and invoke their services.

2.6 Summary

This chapter examines the background of service-based software. It describes the service concept in software, following with service-based software. Service-Oriented Arechitecture (SOA) and Software as a Service (SaaS), as used this thesis, are introduced and compared. An SOA can be used as a software-construction model for constructing software as a service system, while SaaS is employed as a software-delivery model for delivering the software service system as a service to a client. Finally, Web services provide a form of service software technology for developing and deploying the software service system.

Chapter 3

The IBHIS Broker

Chapter 2 examined the background of service-based software and described the service concept in software and software service model including SaaS. This chapter, then, describes the architecture and operation of the IBHIS broker, which is a 'proof of concept' of SaaS. It also decribes how the IBHIS broker was implemented by using Web services.

3.1 Introduction

The Integration Broker for Heterogeneous Information Sources (IBHIS) project was undertaken by the members of the Pennine Group (researchers from the University of Durham, Keele University and UMIST), in partnership with Social Service collaborators in Solihull and Keele's Centre for Health Planning and Management (Budgen *et al.*, 2005, 2007). This project was funded by EPSRC through its Distributed Information Management programme. It aimed to employ a fully service-based approach to support the trustworthy integration of heterogeneous forms of information possessed and manipulated by autonomous agencies.

The IBHIS broker demonstrated the overall set of SaaS concepts in a form that was based on services that deliver data (Budgen *et al.*, 2005). IBHIS explored a service-based approach to large-scale data integration, gathering information from distributed, heterogeneous and autonomous data sources (Budgen *et al.*, 2004). It focused on the needs of healthcare information, particularly health and social care, and demonstrated how data sources as well as their data could be integrated while also being held and managed by disparate and autonomous healthcare agencies. Within the structure of UK health and social care, this could provide an organisation changing context which could then be used by various clients for diverse purposes.

3.2 The Broker Concept

In mercantilism, a broker is a party that arbitrates between buyers and sellers. The concept of the broker refers to someone or something that is responsible for collecting information and blending it into a composite form (Budgen *et al.*, 2005). The broker takes action on behalf of the personal interests of an end user. It merges the requirements and preferences of the end-user based on their own cognition of the 'solution domain'.

The broker generally gathers the information directly from sources and carries out any neccessary tasks. However, it does not demand to take possession of the information received from the sources. The delivery of information, via an information broker, "offers the opportunity of integrating data from a range of autonomous agencies, while at the same time preserving any restrictions on access to or use of the information that either agencies or legal and ethical frameworks might impose" (Budgen *et al.*, 2007).

The IBHIS system has applied the concept of the broker to the use of electronic healthcare records in several aspects, such as finding the appropriate sources and representing the end user's requirements. The IBHIS broker acts as a trusted intermediary gathering, protecting and aggregating information from electronic sources retained in different distributed agencies.

3.3 The Architecture of IBHIS

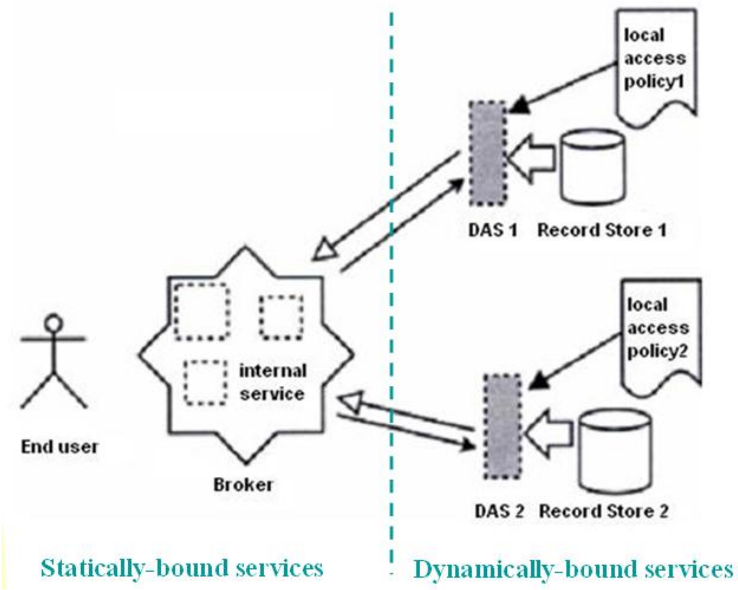


Figure 3.1: A schematic of the IBHIS structure taken from (Budgen et al., 2005)

Figure 3.1 provides a schematic of the IBHIS structure. It is composed from basic services used in two forms (Budgen *et al.*, 2007, 2004; Rigby *et al.*, 2007).

- Statically-bound services

  The statically-bound services are the set of services that are known in advance. They are used within the broker to provide functional tasks such as user interface, access control mechanisms, and query formulation.

- Dynamically-bound services

  The dynamically-bound services are the set of information services that are termed as Data as a Service (DaaS). DaaS is a service supplying information, used where the group of related sources is usually regulated dynamically. Therefore, the dynamically-bound services that are needed to perform a task, are determined, located and bound at the time of execution. They use the concept of a Data Access Service (DAS) to form an interface between the 'physical' structure of the data sources, and the broker itself.

According to a query specified by an end user, the IBHIS broker can dynamically locates and accesses distributed data sources, provides as data services, selecting these according to the information that they can provide.

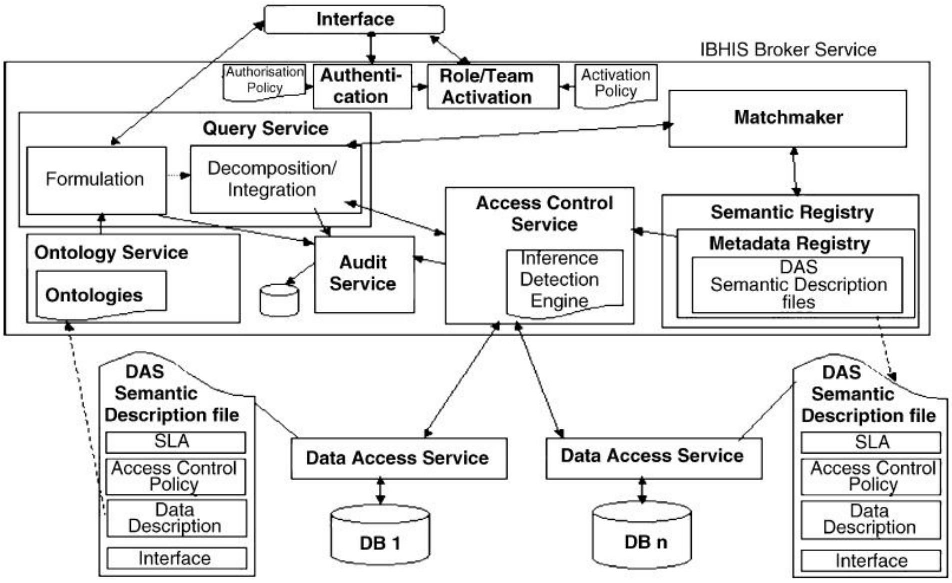


Figure 3.2: The architecture of IBHIS taken from(Rigby et al., 2007)

Figure 3.2 shows the architecture of the IBHIS prototype. It consists of a number of main static service components. These components work together to accomplish the tasks needed for the query of the end user. The service functions of the component are described briefly as follows.

1. *Graphic user interface* (GUI)

   The GUI provides the user interface for the user authentication, in conjunction with the access control service, and also for the query formulation operation that issues the query to the query service component. Finally, after collecting the data, it presents the results to the end user.

2. *Access control service* (ACS)

   The ACS is used for any user authentication needed to access the other system resources. Each resource has its own access rules and security implementations of the data access's authorisation.

3. *Query service* (QS)

   The QS consists of two main components performing the following tasks.

   (a) *Query decomposer* generates a set of local queries that are decomposed from the query. It uses information from the matchmaker and the semantic registry during the query decomposition process.

   (b) *Query integrator* collects the results from the data access services and and integrates them into an appropriate form.

4. *Ontology service* (OS)

   The OS is employed by the QS for the query decomposition and integration process. Queries and responses from the DASs are expressed in terms of the relevant ontology.

5. *Audit service* (AS)

   The AS consists of two components used to retain information about every action of IBHIS, so that the information may be recreated or audited later.

   (a) *User audit* retains information related to the end user, for instance user login and logout date; and the details of queries.

   (b) *System audit* keeps two types of information that are related to data sources and user setup. The examples are intervals of data availability and user registration respectively.

6. *Data access service* (DAS)

   The DAS forms an interface between the information broker and the data sources that are heterogeneous, distributed and autonomous. Section 3.4 will provide more detail about the structure of a DAS.

3.4 The Data Access Service (DAS)

The Data Access Service (DAS) is a major part of the IBHIS architecture (Budgen *et al.*, 2007, 2005; Rigby *et al.*, 2007) (Zhu *et al.*, 2004). It promises as a service-oriented interface for the information broker to transparently access different healthcare data sources which are heterogeneous, distributed, and autonomous. Figure 3.3 shows the architecture of the DAS which consists of four main parts: semantic data description, DAS engine, metadatabase, and data sources.

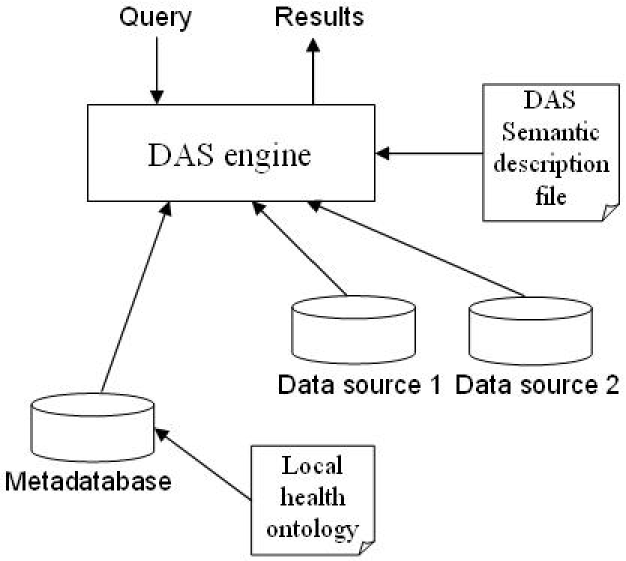


Figure 3.3: The architecture of DAS taken from (Budgen et al., 2005)

1. The *DAS data description* provides a semantic description of the elements of the DAS. The semantic data description is described with an ontology-based approach. Examples of the data description's attributes are the business content (vacabulary), content and data types. Therefore, a service consumer can find and use a service sematically and correctly. The data description provides information arranged into four main types as follows.

- Details of the data, both input and output including its format, for instance the input data consisting of permission identity and assigned roles of the consumers; and the output data consisting of query result. Both of them are represented in XML format.
- The domain and functionality associated with the data items, such as Person domain with GP functionality
- The authorization policies for data usage, for example a hospital patient allowed for querying data
- Other non-functional specifications, such as quality of service and (if appropriate) any fees that might be charged

The DASs then publish the semantic descriptions to the service registry. The IBHIS broker can discover and consume the DASs' services based upon the DASs' semantic description to meet a user's needs.

2. The *DAS engine* is the key element of the DAS. The DAS engine provides data access services for the broker. It analyses and processes queries for authorizing and accessing local data sources, consulting together with metadatabase as necessary.

3. The *Metadatabase* provides for the management of evolution and changes that take place within and outside of an organization. The changes involve the format and structure of data, constraints, authorization or rules, data models and sematics. In particular, it details how the data items of local data sources are mapped by the service provider to the domain ontology. The domain ontology is used as the global schema. The user uses the domain ontology to create a query by navigating an ontology. The ontology is basically a conceptualization of the context where the user is performing. When there are changes needed, the service provider need only alter those local records

relating to the change in the metadatabase. The advantage of this approach is that it can reduce the maintenance overhead for the DAS and the broker.

4. The *Data sources* provide local healthcare data for the broker. They are the local data storage mechanisms that may be distributed and heterogeneous. They may retain data in various forms, for instance relational and object-oriented data models. The detailed storage forms should not be seen by the broker because the DAS description file also has the role of enforcing information hiding. The DAS engine accesses these data sources on a read-only basis.

3.5 Operation of IBHIS

Figure 3.4 shows the main operations in IBHIS. Based upon the details of the request from an end user, the following set of operations or processes are employed to achieve the neccessary tasks for the end user.

1. The "Create Query" is performed by the end user, through the user interface, for user authentication and query formulation.

2. The "Process Query" operation analyses an end user's query and locates candidate DASs that may provide relevant data.

3. The "Attribute Authorisation" process retrieves access permissions from the relevant DASs and matches them with the query.

4. The "Data Access Service" deals with the query submitted by the end user. The query is translated into SQL and executed by the data access services within the selected DASs. The query results are generated in XML format so that they can be returned via the SOAP protocol.

5. A "Content Authorisation" operation is performed to select only the results that the end user is permitted to see.

6. The "Results Processor" arranges the results and presents them to the end user.

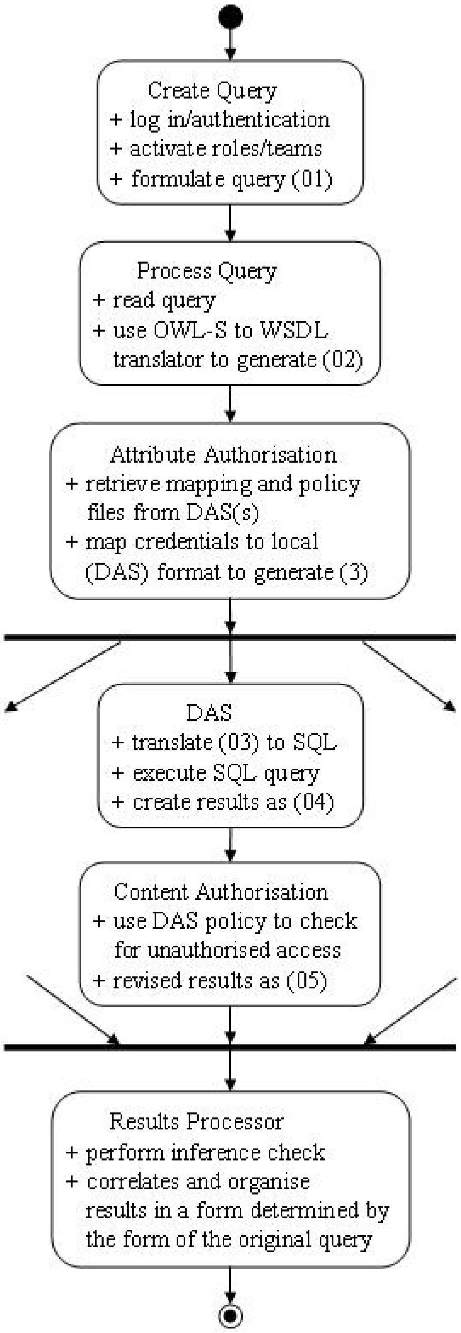

Create Query
+ log in/authentication
+ activate roles/teams
+ formulate query (01)
Process Query
+ read query
+ use OWL-S to WSDL translator to generate (02)
Attribute Authorisation
+ retrieve mapping and policy files from DAS(s)
+ map credentials to local (DAS) format to generate (3)
DAS
+ translate (03) to SQL
+ execute SQL query
+ create results as (04)
Content Authorisation
+ use DAS policy to check for unauthorised access
+ revised results as (05)
Results Processor
+ perform inference check
+ correlates and organise results in a form determined by the form of the original query

Figure 3.4: The processing a query in IBHIS taken from (Rigby *et al.*, 2007)

3.6 Services in IBHIS

IBHIS demonstrates the service-integration layer of SaaS that includes the following five main service functions (Budgen *et al.*, 2007, 2004). However, IBHIS is constrained to using any services that deliver data. Hence, the service model is interpreted as follows:

- Service description includes service characteristics, for instance service interface, functional and non-functional service characteristics as well as constraints. So the end user can match the needs to the service description of available services.
- Service discovery supports the end user by locating appropriate required services from a list of candidate services and providers. Both service description and discovery are employed in accessing data services.
- Service negotiation is mainly employed as the means of providing access-control issues. It is used to verify the local rules, retained in the data services, to ensure that the end user has authorization to access specific data sources.
- Service composition is the process of creating services from a group of lower level services. IBHIS uses service composition to gather and present the information for an end user.
- Service delivery is carried out through both invocation and provision of data services except suspension.

3.7 Implementation

A number of software development tools and environment were used to build the prototype of IBHIS (Budgen *et al.*, 2005; Zhu *et al.*, 2006; Turner *et al.*, 2004).

1. The prototype was built by using IBM's WebSphere Application Server v5.1 working on the Windows, Linux and Solaris operating system. The IBM WebSphere provided development and deployment tools for implementing, testing and deploying web services that were based on open, cross-platform standards: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language) and UDDI (Universal Description Discovery and Integration).

2. The user interface of the IBHIS broker was implemented by using Java JSP (JavaServer Pages), Java servlets and HTML (HyperText Markup Language) pages.

3. IBHIS provides a user interface for the ontological query formulation whereby an end user uses a healthcare ontology to navigate using familiar terms and results in order to formulate a query. This user interface was developed by using Jena v2, including the inbuilt query language RD-QL (RDF Data Query Language). The RDF (Resource Description Framework) is a standard model that is used for data interchange on the Web. RD-QL is used during the formulation process to manipulate and search the ontology.

4. The DASs implemented and described services by employing service description language, for instance WSDL and DAML-S. The DASs published their service descriptions in a semantic registry.

5. The DASs were deployed on different relational database management systems (RDBMS) including MySQL, IBM DB2, Oracle and Microsoft Access. The RDBMSs were implemented on heterogenous platforms at distributed sites

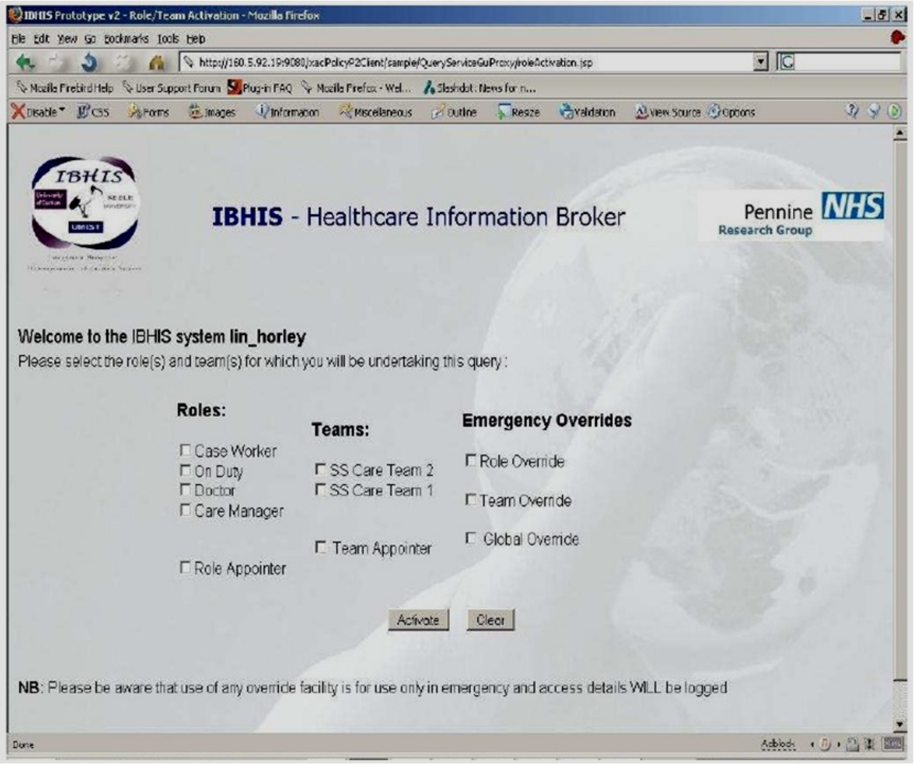
6. The semantic registry was based on UDDI, used to hold semantic descriptions and handle semantic queries.

7. The ontology service was implemented using the OWL (Web Ontology Language) to manage an ontology of healthcare terms.

8. The messages were exchanged within the IBHIS system as SOAP document style messages.

Figure 3.5 shows the user interface of IBHIS role/team activation. A user uses the user interface to access to the IBHIS system by choosing a combination of roles and teams. Then they are presented with the query formulation screens, as shown in Figure 3.6.

The query formulated by the user, as shown in Figure 3.6, was added to the XML document. The query was then sent as a SOAP document style message to the IBHIS broker's 'Query Service' in order to locate and invoke the relevant DASs' data service for the user. Figure 3.7 illustrates an example of the result from using IBHIS.


IBHIS Prototype v2 - Role/Team Activation - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://160.5.92.19:9080/xacPolicyP2Client/sample/QueryServiceGuProxy/roleActivation.jsp
Mozilla Firebird Help
User Support Forum
Plug-in FAQ
Mozilla Firefox - Wel...
Slashdot: News for n...
Disable
CSS
Forms
Images
Information
Miscellaneous
Outline
Resize
Validation
View Source
Options
IBHIS
Pennine NHS Research Group
Done
Adblock

IBHIS - Healthcare Information Broker

Welcome to the IBHIS system lin_horley

Please select the role(s) and team(s) for which you will be undertaking this query :

Roles:

☐ Case Worker
☐ On Duty
☐ Doctor
☐ Care Manager
☐ Role Appointer

Teams:

☐ SS Care Team 2
☐ SS Care Team 1
☐ Team Appointer

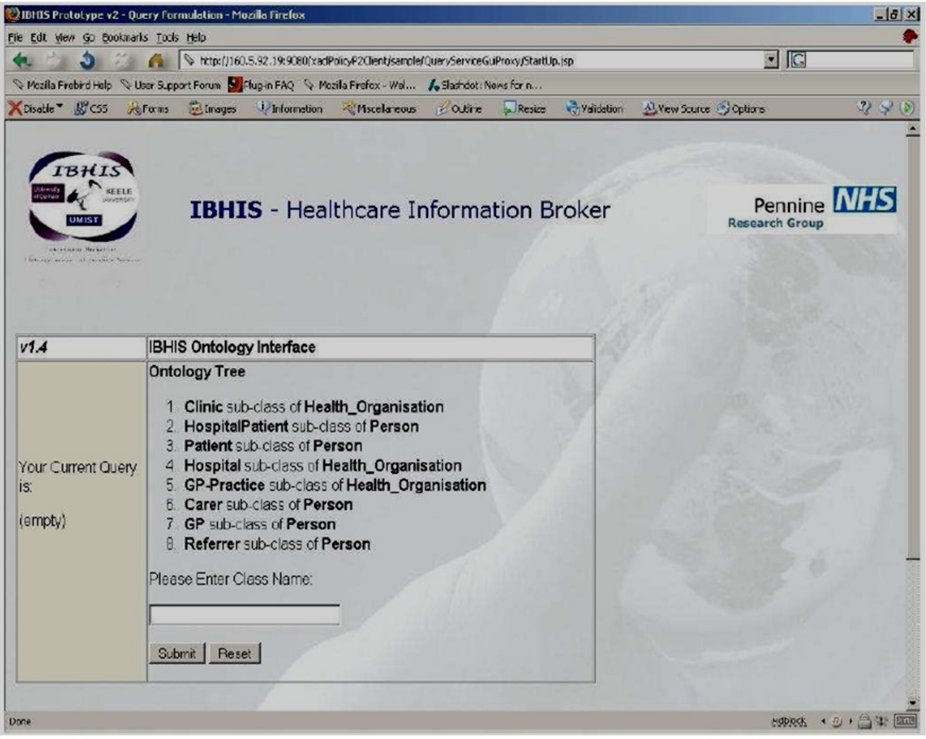
Emergency Overrides

☐ Role Override
☐ Team Override
☐ Global Override

Activate Clear

NB: Please be aware that use of any override facility is for use only in emergency and access details WILL be logged

Figure 3.5: IBHIS role/team activation screen



IBHIS - Healthcare Information Broker

Pennine NHS Research Group

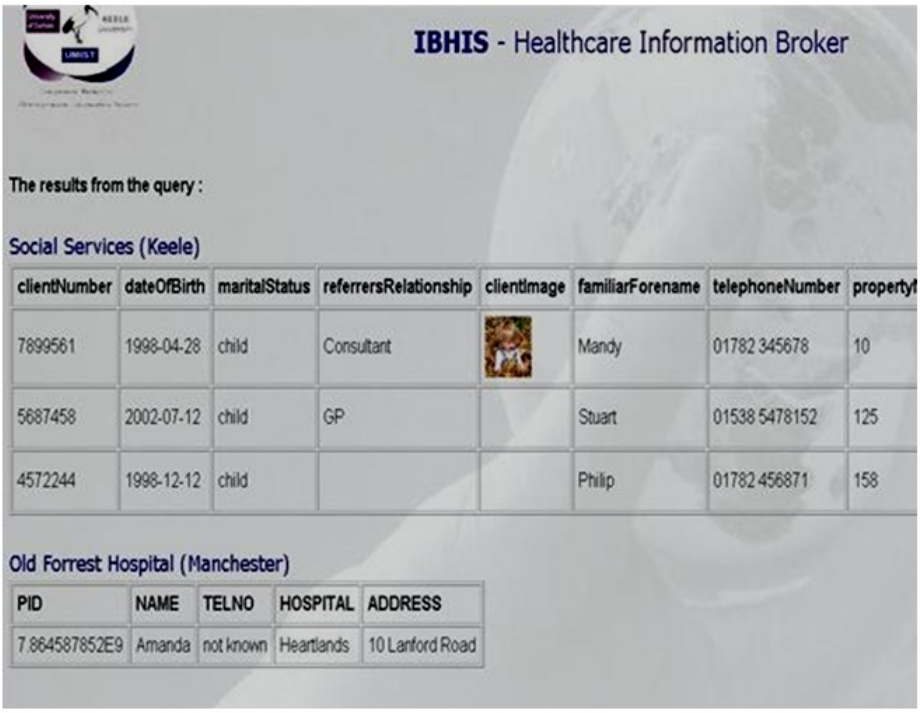
v1.4	IBHIS Ontology Interface
Your Current Query is: (empty)	Ontology Tree

1. **Clinic** sub-class of **Health_Organisation**
2. **HospitalPatient** sub-class of **Person**
3. **Patient** sub-class of **Person**
4. **Hospital** sub-class of **Health_Organisation**
5. **GP-Practice** sub-class of **Health_Organisation**
6. **Carer** sub-class of **Person**
7. **GP** sub-class of **Person**
8. **Referrer** sub-class of **Person**

Please Enter Class Name:

Submit Reset

Figure 3.6: Ontological query formulation screen



IBHIS - Healthcare Information Broker

The results from the query :

Social Services (Keele)

clientNumber	dateOfBirth	maritalStatus	referrersRelationship	clientImage	familiarForename	telephoneNumber	property
7899561	1998-04-28	child	Consultant		Mandy	01782 345678	10
5687458	2002-07-12	child	GP		Stuart	01538 5478152	125
4572244	1998-12-12	child			Philip	01782 456871	158

Old Forrest Hospital (Manchester)

PID	NAME	TELNO	HOSPITAL	ADDRESS
7.864587852E9	Amanda	not known	Heartlands	10 Lanford Road

Figure 3.7: The result from IBHIS's query

3.8 Summary

This chapter describes the architecture and operation of the IBHIS broker. It is a ‘proof of concept’ of SaaS providing a set of SaaS’s service functions in a form that is based on services that deliver data. The architecture of IBHIS is composed from basic services used in two forms: statically-bound and dynamically-bound services. The statically-bound services are the set of services that are known in advance. They are used within the information broker of the IBHIS to provide functional tasks such as query formulation. The dynamically-bound services are the set of information services termed as Data as a Service (DaaS). The DaaS is as a service-oriented interface for the information broker to transparently access different data sources which are heterogeneous, distributed and autonomous. The prototype of IBHIS was built and run based on Web Service technologies. The prototype support Web Services standards, for instance SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration) and XML (Extensible Markup Language).

Chapter 4

Negotiation

Chapters 2 and 3 examine the background of Service-based software and the IBHIS broker respectively. In addition, this chapter examines the background of negotiation from various perspectives. It describes the principle of negotiation as well as the types of negotiation studied and applied in different aspects. Then automated forms of negotiation are discussed and we introduced some negotiation models that are useful for this research.

4.1 Negotiation Fundamentals

Negotiation is a widely-used process, not only for the way that people interact or deal with others, but also now people work and live according to the new era of computer technology. This section will describe the fundamentals of negotiation from various aspects of studies in negotiation. It introduces, for instance, the definition of negotiation, its characteristics, and negotiation suitiations. Then the types of negotiation are described as well as the models used for automated negotiation.

4.1.1 The Nature of Negotiation

Negotiation has been studied from various perspectives as illustrated in Figure 4.1. Most negotiation models and systems in computer science and information systems are based on the negotiation procedures and models taken from economic science and management as well as those used in law and the social sciences (Bichler et al., 2003). Nevertheless, computational models and systems have led to an increasing number of negotiation techniques, models and procedures.

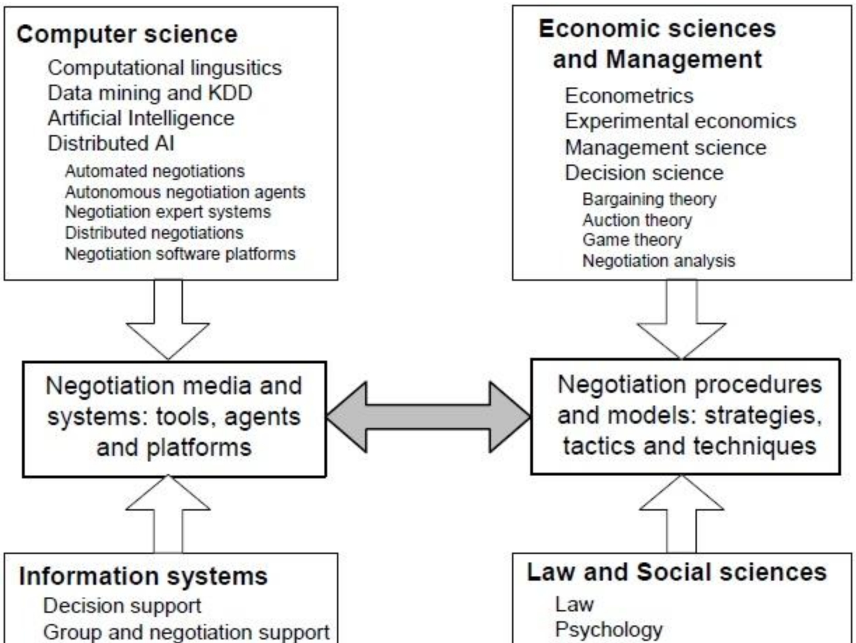


Figure 4.1: Negotiation research areas, their results and key influences taken from (Bichler et al., 2003)

These research areas for negotiation are described as in the following (Bichler et al., 2003).

1. Law and social sciences place emphsis upon “the prescriptive and descriptive models, heuristics and qualitative studies of negotiation processes and

negotiators' behaviour".

2. Economics and management science address "the construction of formal models and procedures of negotiations, rational strategies and the prediction of outcomes".

3. Computer science and information systems focus on the "construction of electronic negotation tables, decision and negotiation support systems (DSS, NSS), artificial negotiating software agents (NSA) and software platforms for bidding and auctioning".

Various definitions of negotiation have been proposed from different aspects, such as electronic commerce, social science and artificial intelligence.

- *Social science*: "The process by which group of agents communicate with one another to try and come to a mutually acceptable agreement on some matter" (Lewicki et al., 2006).

- *Electronic commerce*: "The process by which a group of agents communicate with one another to try to reach agreement on some matter of common interest" (Lomuscio et al., 2003).

- *Software as a service (SaaS)*: "A process of interaction between the client and one or more service providers, with the aim of agreeing the terms and conditions for the supply of a service" (Budgen et al., 2004).

Therefore, when there is a conflict of interest between or among negotiation participants, negotiation is needed to resolve their conflict. The process of 'give-and-take' is the "heart of negotiation" to meet their agreement (Lewicki et al., 2006). The need for negotiation may arise for both functional and non-functional issues (Dang and Huhns, 2005).

Negotiation can be categorised as follows (Raiffa, 1982):

- *Bilateral* negotiation is one-to-one negotiation in which two parties negotiate with each other.
- *Auction* negotiation is one-to-many negotiation where one party, an auctioneer, negotiates with a number of other parties, the bidders.
- *Distributed and multilateral* negotiation is many-to-many negotiation where the parties involved negotiate for agreements.

In addition, negotiation can be considered based on the number of parties, the number of issues, and the negotiation situation disputed. Therefore, negotiation is classified with this consideration into three kinds: two parties/one issue, two parties/many issues, and many parties/many issues (Raiffa, 1982).

Negotiation in different situations needs different negotiation protocols, so that the participants can negotiate to meet their objectives (Bichler, 2000*b*). However, the participants have their own reasons for negotiation as follows (Bichler et al., 2003).

- Cannot achieve their objectives through unilateral actions
- Exchange information comprising offers, counter-offers and arguments
- Deal with interdependent tasks
- Search for a consensus which is a compromise decision”

4.1.2 Distributive and Integrative Negotiations

Traditionally, interdependent negotiation situations can be classified into two types: zero-sum, or distributive situations; and non-zero-sum, or integrative or mutual gains situations (Hung et al., 2004; Lewicki et al., 2006).

- A distributive situation (win-lose) arises where the parties are trying to get the larger share of a limited resource.

- An integrative situation (win-win) is required by the negotiators to search for solutions that will benefit both parties.

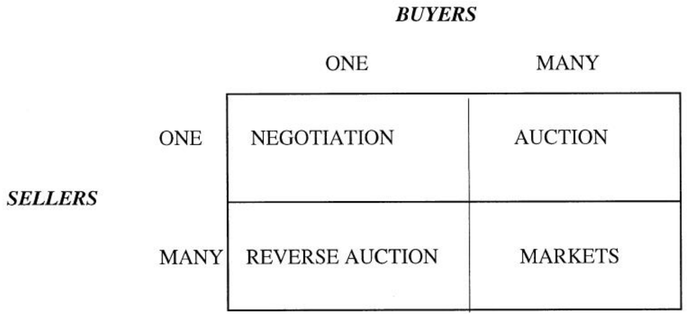
Negotiation can also be classified as being cooperative or non-cooperative in form. Beam and Segev state that "Cooperative negotiation assumes, for the most part, that all parties are working towards fundamentally the same goal." (Beam and Segev, 1996). Nwana also observes that collaborative agents place stress on autonomy and collaboration with other agents to operate tasks for their owners (Nwana, 1996). Negotiation may be needed to accomplish mutually acceptable agreements on some issues.

4.2 Types of Negotiation

Negotiation has particularly been studied in three main areas: auction, game theory and bargaining. Auctions have been used widely especially in electronic commerce for the last recent years. Game theory is a branch of applied mathematics that is used in various aspects, particularly economics and social science. Bargaining is widely used in trading between buyers and sellers to negotiate the price for goods and services that they have to pay and sell. This section will examine and describe these three main types of negotiation.

4.2.1 Auction

"Auctions are market mechanisms with an explicit set of rules determining resource allocation and prices based on bids from the market participants" (cited in Gimpel et al., 2008). As shown in Figure 4.2, auctions involve in trading by many sellers and buyers in the market framework. There are three general types of activities associated with an auction: "receiving bids, clearing and revealing intermediate information" (Ströbel and Weinhardt, 2003).



		BUYERS	
		ONE	MANY
SELLERS	ONE	NEGOTIATION	AUCTION
	MANY	REVERSE AUCTION	MARKETS

Figure 4.2: Market framework taken from (Teich et al., 1999)

Auctions can be classified into two main groups: open and sealed-bid auctions (Bichler, 2000*a*; Li, 2001; Teich et al., 2004).

1. Open auction

   Open-cry English and Dutch auctions are two popular open auctions. Auctioneers of both groups reveal price quotes in which prices go up and down and the winners are the highest and lowest price bidders respectively.

2. Sealed-bid auction

   The sealed-bid auction or a Vickrey auction propose the price without revealing it to other people. The winner is the highest bidder, but the price is paid at the second-highest bid. These are the single-sided auctions. Only the bidders bid their price quotes. But double-sided auctions allow a number of buyers and sellers simultaneously submit bids at each auction round.

The multi-attribute auction is a particular type of auction that is mostly deployed in a specific negotiation situation, the so-called monopsony situation, such as corporate procurement (Bichler et al., 1999; Bichler, 2000*a*). It supports a multi-lateral automated negotiation situation in which negotiation is based on multiple attributes of a deal. For the multi-attribute reverse auction, a buyer acts as an auctioneer rather than a bidder, which is a role played by multiple suppliers (David et al., 2006). The buyer solicits multi-attribute bids of products and services from a group of potential suppliers. The winning supplier is the one who offers the bid giving the highest utility of all attributes for the buyer and best fulfils the buyer's preferences. The bid is generally organized as a set of attribute-value pairs (Bichler and Kalagnanam, 2005). The attributes are based on quantitative and qualitative features such as price and quality of service. According to an experimental study comparing the overall utility achieved, the multi-attribute auction mechanisms are significantly better than single attribute auction mechanisms (Bichler, 2000*a*). Hence a bid-taker in the multi-attribute auction profits more than the one in the single attribute auctions.

The general approach to eliciting and assessing the buyer's preferences is based on multi-objective decision analysis which quantitatively analyzes imperative decisions relating to multiple and interdependent objectives (Bichler et al., 1999; Bichler and Kalagnanam, 2005). The common decision analysis techniques are MAUT (Multi-Attribute Utility Theory), AHP (Analytic Hierarchy Process) and conjoint analysis which are used to determine the utility function of the buyer. MAUT focuses on the assessment of the individual utility functions and weights. It uses utility functions to evaluate quantitative decision problems analysing preferences with multiple attributes (Guttman & Maes, 1998). Therefore the buyer can compare the multi-attribute bids or the set of feasible alternatives, determine differences among them and, then, select the bid that is given the highest overall utility rating (Bichler et al., 1999).

Characteristic	Traditional auctions	Traditional negotiations	On-line auctions
Number of participants	Multi-bilateral, single or double-sided	Bilateral, multilateral or multi-bilateral; arbitrary number of sides	Multi-bilateral, single or double sided
Participants	Open or restricted	Restricted	Open, restricted or rule-defined
Consensus required	Bi-taker and selected bidder	Selected or for all participants	Selected participants
Number of objects	Single, homogenous	Single or multiple, homo- or heterogeneous	Single or multiple homo- or heterogeneous
Number of issues	Single	Single or multiple	Single or multiple
Issues structure	Well-defined	Well-defined, partially, or ill-defined	Well-defined
Offer space	Fixed	May be unknown and modified	Fixed
Exchange and knowledge of offers and concession-making	Yes	Yes	Yes
Logrolling (conditional concessions)	No	Yes	Yes
Knowledge of offers and concessions)	Public or private	Private (rarely public)	Public or private
Exchange of opinions, arguments, threats	No	Yes	No
Interdependence	Between bid-taker and bidders (single sided) or between but not within sides (double-sided	Full interdependence except multi-bilateral negotiations	Between bid-taker and bidders (single-sided) or between but not within sides (double-sided)
Protocol	A priori defined, explicit and fixed	Well-defined or partially defined; explicit or implicit	Apriori defined, explicit and fixed
Competition versus Cooperation	Competition among bidders on at least one of the possibly two sides: cooperation prohibited	Competition or cooperation among the agent	Competition among bidders on at least one of possibly the two sides; cooperation prohibited
Process control	Defined a priori	Ill-defined, modifiable by participants	Defined a priori

Table 4.1: Characteristics of negotiations and auctions taken from (Bichler et al., 2003)

Traditionally auctions have some different characteristics from negotiations, as shown in Figure 4.1 (Gimpel et al., 2008). Negotiations are perceived as a cooperative process that resolves multiple issues, whereas auctions involve competitive processes solving problems with a single issue. In addition, negotiations and auctions involve bilateral and multi-lateral interaction of paticipants respectively. However, on-line auctions and traditional negotiations are similar because of many main characteristics, especially multi-bilateral negotiation, competitive bidding process and the number of issues (Bichler et al., 2003). Table 4.1 provides more details about the characteristics of negotiations and auctions.

4.2.2 Game Theory

For decades, models of negotiation process have been developed by using the ideas of game theory (Fatima et al., 2005). The objective of game theory is to examine the players' best options in attempting to maximize results (Faratin, 2000). There are seven elements of a game: players, actions, information, strategies, payoffs, outcomes and equilibrium. A player chooses a strategy to respond according to the rules of a game. The chosen strategy results in a payoff. Game theory is focused on the benefit of the individual instead of the group. It can be used to design a negotiation mechanism and particularly the definition of protocols that restrict the number of strategies used by parties (Sierra et al., 1997). Negotiations such as bargaining decisions and auctions use game theory to analyse their strategic decisions for the prediction of specific outcomes for a certain circumstances (Bichler et al., 2003).

Coordination in game theory can be examined from two aspects: cooperative and non-cooperative (Faratin, 2000).

1. Cooperative games suppose that binding agreements are made in order to coordinate their strategies. Pre-play negotiations may possibly be taken so that a combined course of actions is arranged for the resulting game.

2. Non-cooperative games has mistrust among players of a game. They play the game to maximize their own benefit. Non-cooperative game models are also acknowledged as strategic bargaining theories, where the bargaining situation is modelled as a game. A solution in game theory typically aims at that players' equilibrium strategies. One player's strategy is the most opposite response to the other player's strategy and vice versa.

Games can be classified into four classes (Gibbons, 1992):

- Static games of complete information
- Dynamic games of complete information
- Static games of incomplete information
- Dynamic games of incomplete information .

In a game of incomplete information, a player does not know another player's payoff. The example of this situation is an auction where a bidder does not know how much another bidder will offer for the good being sold. These four classes of games are corresponded to four notions of equilibrium in games: Nash equilibrium, subgame-perfect Nash equilibrium, Bayesian Nash equilibrium, and perfect Bayesian equilibrium.

4.2.3 Bargaining

Bargaining is a process whereby the parties on their own make offers and counter offers to each other in order to come to an agreement (Muthoo, 1999). Unlike auctions, the parties in a bargaining situation have to cooperate to resolve conflict of interests in order to reach a common interest. "A bargaining situation is a game situation in a sense that the outcome of bargaining depends on both players' bargaining strategies" (Muthoo, 1999). Auction and bargaining are considerably different.

Only one side of parties can offer a bid in an auction. In bargaining, many issues can be negotiated (Li, 2001).

Gerding et al. have presented an approach for one-to-many bargaining within the context of agent-mediated electronic commerce (Gerding et al., 2006). A seller and many buyers individually negotiate over multiple interdependent attributes. According to their one-to-many bilateral bargaining protocol, the seller agent negotiates with server buyer agents concurrently. They alter offers and counter offers to each other. Figure 4.3 shows the sequence diagram of the alternating offer bargaining protocol. However, in such one-to-many bargaining setting, customers may be unsatisfied if they perceive that the outcomes of the negotiation are not fair. Thus bargaining strategies for the seller agent are developed to maximize overall revenue by utilizing differences in customers' preferences without violating the fairness constraint. Buyers can be treated similarly. Nevertheless, this research addresses strategies that can employ differences in buyers' time-pressure. The strategies developed determine three aspects.

1. The first aspect is threshold strategies which are used, evaluated and compared by a seller agent. There are five different threshold strategy types: fixed threshold, time-dependent threshold, responsive, responsive with fixed reservation value, and responsive with time-dependent reservation value.

2. The second aspect is multi-attribute offers with a utility level which corresponds to the threshold.

3. The third aspect is a scheme for determining when to respond.

4.3 Automated Negotiation

Research in automated negotiation has been studied from various perspectives (Bichler et al., 2003) for instance, service-oriented architecture (SOA) (Elfatatry and

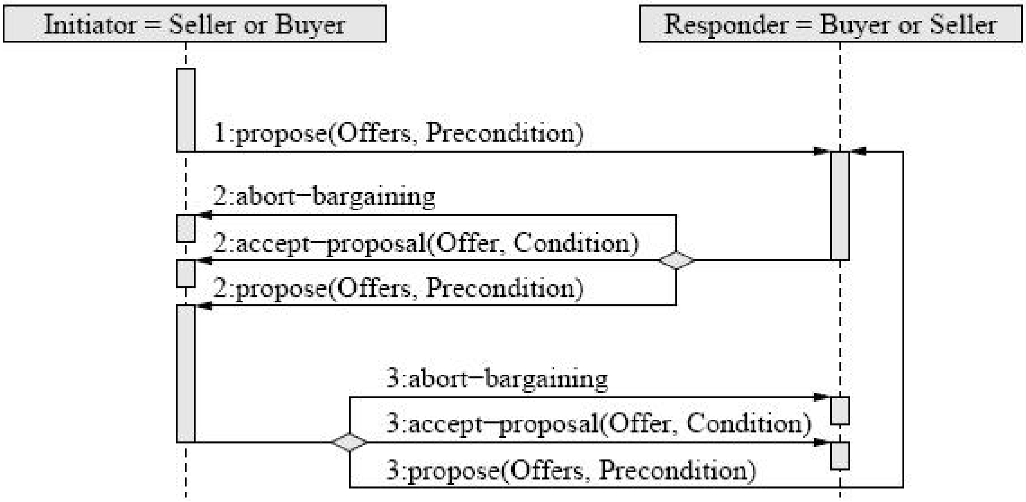

Initiator = Seller or Buyer
Responder = Buyer or Seller
1:propose(Offers, Precondition)
2:abort−bargaining
2:accept−proposal(Offer, Condition)
2:propose(Offers, Precondition)
3:abort−bargaining
3:accept−proposal(Offer, Condition)
3:propose(Offers, Precondition)

Figure 4.3: The agents' bargaining protocol taken from (Gerding et al., 2006)

Layzell, 2004), E-business (Kim and Segev, 2003; Li, 2001; Lomuscio et al., 2003; Strecker, 2004), and artificial intelligence (Dang and Huhns, 2005; Fatima et al., 2004*a*; Jennings, 2000; Carles et al., 1998). Electronic negotiation provides advantages for individuals and organizations from several perspectives. It "promises higher levels of process efficiency and effectiveness, and most importantly, a higher quality and faster emergence of agreements" (Bichler et al., 2003).

Conventionally a model of automated negotiation is based on three fundamental elements: the negotiation object, negotiation protocol and decision making model (Jennings et al., 2001).

1. The negotiation object contains the information of the issues and the aims about negotiation. Negotiation participants exchange a number of negotiation objects during the negotiation process. The negotiation objects contain information that may be used for the negotiation participants' decision-making, based on their conditions and in response to the opponent's offers.

2. The negotiation protocol defines a group of rules regulating the interaction among the negotiation participants involved in a negotiation (Kim and Segev, 2003). For a given negotiation situation, the negotiation protocol must

address a number of negotiation attributes such as the type and number of participants, and item characteristics (Bichler et al., 1999). Therefore negotiation situations which have different solutions need different negotiation approaches to solve the possible conflicts of interest among negotiation participants.

3. The decision making model is used by negotiation participants along with the negotiation protocol to accomplish negotiation goals (Jennings et al., 2001). Each negotiation participant has his/her own decision model which is based on a local cost-benefit model or other strategies (Hung et al., 2004).

Different activities making use of negotiation, such as SOA and electronic commerce, have different characteristics, such as the number of participants, protocols, rules and negotiation environments. Hence, the conventional negotiation model needs to be designed or adapted appropriately to deal with different negotiation circumstances, for instance, service-oriented architecture, E-business, web services, and artificial intelligent. This section examines and describes some automatic negotiation models from these aspects. It focuses on the features and characteristics of the negotiation models that will be useful for designing and developing a negotiation model for this research project.

4.3.1 Service-Oriented Architecture

Elfatatry and Layzell proposed a framework of negotiating in a service-oriented environment as shown in figure 4.4 (Elfatatry and Layzell, 2004). It is for service-oriented software delivering software as a service that needs automated negotiation to tailor needs dynamically. It is defined by a three-phase negotiation process: pre-negotiation, negotiation and service delivery.

1. The *prenegotiation* phase produces the initial information for the negotiation

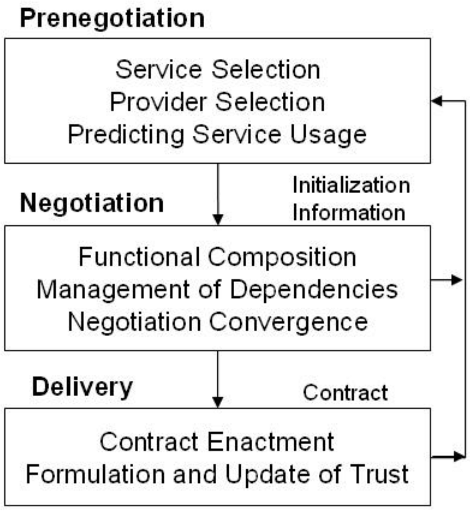


Figure 4.4: The three phases of negotiating in a service-oriented environment taken from (Elfatatry and Layzell, 2004)

phase. It involves with three issues: service selection, provider selection and predicting service usage, as the following steps.

(a) Services are selected by negotiating with service providers offering services with different non-functional issues.

(b) The service providers are selected from a group of service providers. The consumer may retain a list of preferred providers so that the list can be updated with the feedback from the negotiation and the delivery phases.

(c) The service is invoked based on the prediction of service usage. The temporary contracts between the service consumer and provider may be created in advance to provide the predicted method of the service usage if and when the concerned service is invoked. This approach can improve the performance of the system because it reduces the communication between the consumer and provider.

2. The *negotiation* phase involves negotiation participants exchanging their messages to accomplish an mutual agreement by working on three main stages as

follows.

(a) The functional composition stage involves matching functionality from a set of software services. This stage provides the negotiation for the appropriate functional combination needed in order to meet the desired requirements.

(b) Management of dependencies and uncertainty is needed to compose a complex service through negotiation. Negotiation convergence can be achieved with two negotiation mechanisms: negotiation protocols and negotiation strategies, especially in an environment where time is crucial, and negotiation must be taken in a short period of time. Negotiation protocols adjust the behavior of participating parties to minimize time-wasting activities.

(c) To accomplish better convergence, the negotiation strategies approach can employ two techniques: the heuristic approach using different learning techniques and the argumentation approach using techniques that can result the state of mind of the other participant so that a party accepts the proposed offer.

3. Service *delivery* involves implementing the contract agreed by parties. The fulfillment of commitments is investigated and trust values are updated. The results will be kept in the agent's knowledge repository as feedback information to be considered for the next time, if the agent again wants to engage in negotiation with the same partner.

4.3.2 Electronic Commerce

Kim and Segev proposed a negotiation framework for dynamic E-Business negotiation processes (Kim and Segev, 2003). It is a framework for multi-party negotiation which supports a wide range of negotiations in a dynamic e-Business environment. Their motivation was that personalized products and services had been increasingly vital in the dynamic customer-oriented business open market. Consequently, automated negotiation processes are the centre of dynamic interoperable e-Business in the open environment. The framework proposed provides the basis for constructing dynamic negotiation processes by allowing the required functionality in five connected components. Figure 4.5 shows the components of this framework and relationships between them: negotiation requirements, negotiation structure, negotiation process, negotiation protocol, and negotiation strategy.

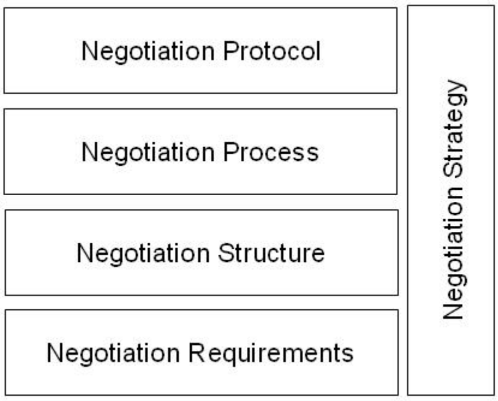


Figure 4.5: The components in the negotiation framework and relationships between them taken from (Kim and Segev, 2003)

1. The negotiation requirements define the issues to be negotiated and the goals to be achieved.

2. The negotiation structure identifies the skeleton of the negotiation describing the structure of a negotiation at different abstract level: role-level structure and partner-level structure. It defines the participants and their relationship involved in the negotiation.

3. The negotiation process component provides the classification of wide ranging negotiation processes used as the basis of design and implementation of negotiation system modules.

4. The negotiation protocol component is the core of negotiation automation providing the rules governed the actions in a negotiation.

5. The negotiation strategy encompasses all aspects of decision making in the negotiation, related to all the other negotiation components, especially the negotiation protocol. Strategic decision making is used to choose an appropriate protocol from a set of protocols. Then the negotiation participant follows the chosen protocol to negotiate with negotiation opponents in order to gain the best deal from them.

In addition, Li (2001) proposed a negotiation model for automated e-business negotiation. It is used to support the inherent complexity of business negotiations. It captures the main concepts and elements related in automated negotiations by defining a seven-tuple <C, AN, M, PT, RCSM, CBESM, DM>.

- C stands for clients representing negotiation participants from different business enterprises.

- AN stands for automated negotiators carrying out negotiation on behalf of its clients.

- M means a group of messages holding meaningful negotiation information that express the purposes of a negotiation sender as well as describing data and constraints. The messages are exchanged among negotiation participants. When a sender sends the message to a receiver, the receiver then analyses the message in order to respond appropriately to the sender.

- PT represents negotiation protocol identifying a finite state which consists of a number of states and transitions (alternative actions) of a negotiation process. Automated negotiators must follow negotiation protocol during negotiation process to reach an agreement.

- RCSM is used for a requirement and constraint specification model representing the needs or interests of automated negotiators (AN).
- CBESM denotes a Cost Benefit Evaluation and Selection Model. It is used by automated negotiators during negotiation process to evaluate and select the best option in terms of their cost and benefits.
- DM denotes a decision model. It is an important tuple needed to capture a business enterprise's negotiation concepts: goals, policies, strategies, plans of decisions and actions and their inter-relationships.

Figure 4.6 shows the inter-relationship of key concepts in the decision model.

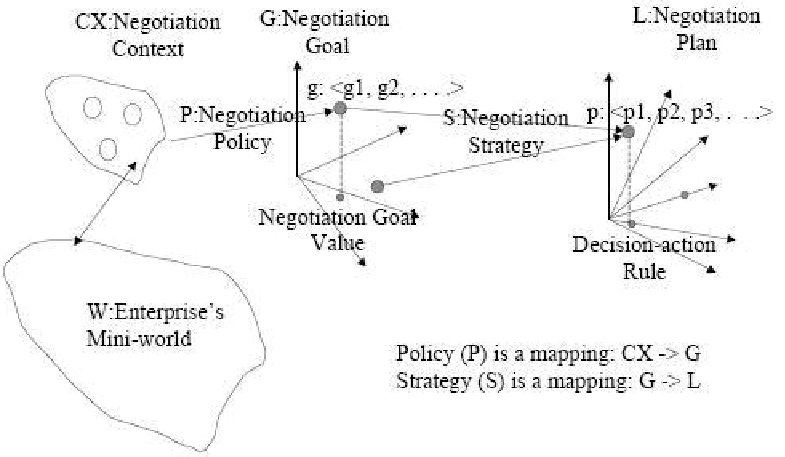


Figure 4.6: Key concepts in the Decision Model taken from (Li, 2001)

The decision model is specified as a six-tuple <W, CX, G, L P, S>.

- W represents an enterprise's mini-world. They are resources, such as information, materials, and personnel, accessed by the enterprise.
- CX denotes a set of negotiation context which is defined by the enterprise.
- G represents a group of negotiation goals of the enterprise.
- L represents a set of plans of decision and actions. This set is employed by an automated negotiator.

- P is negotiation policy mapping a negotiation context to a negotiation goal.
- S denotes for negotiation strategy. It maps negotiation goals to the plans of decisions or actions.

During a negotiation process, a well-defined protocol, is followed by automated negotiators to exchange negotiation primitives and data. A negotiation protocol can be defined by a finite state diagram that is a type of diagram used to describe the behavior of systems. The diagram is composed from a number of states and transitions. At each state, a decision or an action needed by an automated negotiator is made or taken based on some conditions before the subsequent state transits. A decision-action rule can be formed from the specification of that conditional decision or action. The example of the transition in different states of the bi-lateral negotiation protocol are the initiation of a negotiation transaction, the acceptance/rejection/modification/withdrawal of a transaction, the termination of a negotiation transaction, and the generation of a counterproposal.

4.3.3 Web Services

Hung et al. introduce WS-Negotiation for Web services as “a bilateral and multi-issue bargaining Web services language” (Hung et al., 2004). It is an independent declarative XML language used by Web services providers and requestors to negotiate and find an acceptable agreement on both sides.

Figure 4.7 illustrates the structure of WS-Negotiation mapping with negotiation entities. It consists of three main elements: negotiation message, negotiation protocol and negotiation decision making .

- Negotiation message is the description of the negotiation message involving a set of issues that holds a set of alternatives. It is exchanged among negotiation participants during negotiation process. There are many kinds of negotiation

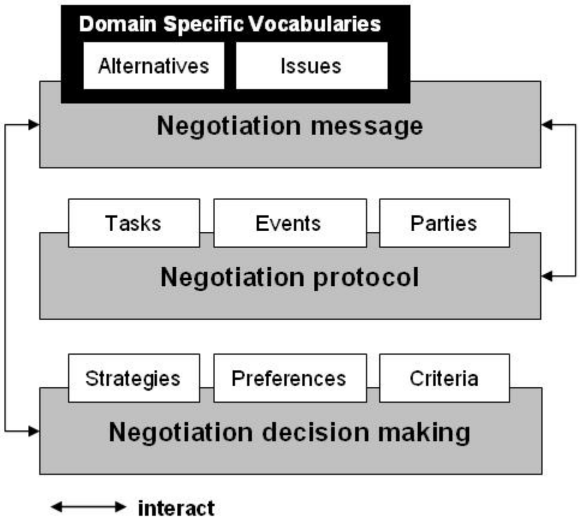


Figure 4.7: The structure of WS-Negotiation taken from (Hung et al., 2004)

messages, for instance, offer, counter-offer, accepted, rejected, expired, signed and unsigned.

- Negotiation protocol identifies the mechanism and rules followed by the negotiating participants or parties. The participant employs several tasks and activities to communicate and cooperate among them, such as Call For Proposal (CFP), propose, accept, terminate, reject and acknowledge.

- Negotiation decision making is an private decision process. The negotiation participant has its own strategy for generating a set of preferences for the decision-making process, to deal with an offer or counter-offer, and to achieve an agreement in the negotiation protocol. The group of preferences are related to the alternatives of each issues of the negotiation message.

Figure 4.8 shows the relationships between the negotiation entities that are mapped to the negotiation elements of WS-Negotiation.

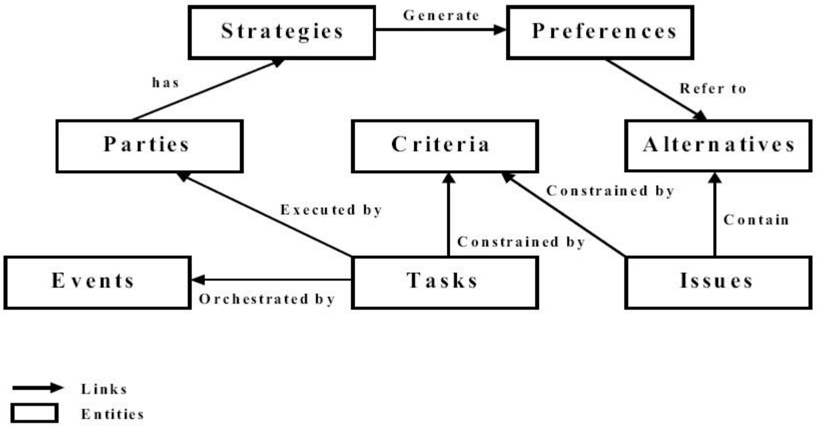


Figure 4.8: Relationship of negotiation entities taken from (Hung et al., 2004)

4.3.4 Artificial Intelligence

Dang and Huhns propose a negotiation model for coalition deal negotiation for services, aiming to "optimize the agents' utilities with minimized computational cost" (Dang and Huhns, 2005). The negotiation among agents (both service requestors and providers) is based on functionality and quality of service (QoS) issues. They employ an optimal strategy for multiple-issue negotiation to achieve a service agreement. The negotiation model is composed of four key elements:

- Negotiation set defines a set of possible proposal value of negotiation issues that are based on functionality and QoS.
- A protocol specifies "the legal proposals that an agent can make".
- A strategy is used by the agent to make the proposals. The decision-making involves the agent's preferences.
- A rule is governed by a mediator to identify when to propose a deal and what the agreement is.

Fatima et al. (2004*a*)introduce an agenda-based framework for multi-issue negotiation under time constraints in an incomplete information setting . The purpose of this model is to be closer to an almost real-practice bargaining situation. The negotiation model consists of four components:

- The negotiation protocol identifies the rules of encounter within the circumstances in which the agents interact with each other.
- The negotiation strategy is a specific plan of the sequence of actions to be followed by the agents during negotiation.
- The information state of agents contains details about the private information that each agent has, and that is not known to its opponents, according to the negotiation game.
- The negotiation equilibrium is an equilibrium strategy used by the agent to choose a best response to its opponent's strategy during negotiation game. It is different from a negotiation mechanism that is composed from the negotiation protocol and negotiation strategies. The negotiation mechanism is static because it deals with the certain situation.

4.4 Summary

This chapter examines the background of negotiation from several points of view. Negotiation is a process of interaction between or among negotiation participants to resolve their conflict. There are three common types of negotiation: auction, game theory and bargaining. An auction is a process of trading goods or services by many sellers and buyers in the market framework that the goods or services are sold to the highest bidder. Game theory focuses on the ways in which strategic interactions among parties produce outcomes with respect to those parties' preferences. Bargaining is a process whereby the parties have to cooperate to resolve conflict of interest in order to reach a mutual interest.

Automated negotiation provides more efficiency and effectiveness in the negotiation process than human negotiation process. The automated negotiation is based on three fundamental elements: negotiation object, negotiation protocol and decision model. In addition, the traditional negotiation model could be designed and developed appropriately to fit with the different context or situation.

Chapter 5

The Integrated Care Plan

This chapter explores how both the concepts and role of the broker as well as of service negotiation have been extended from those used in IBHIS.

5.1 Introduction

Within the healthcare sector, an integrated care plan is a tool used to manage a patient's case, in the situation where a potentially complex set of health and social care resources are needed over a period of time. It provides details about the roles and responsibilities of care from those professionals who provide hospital and social care treatments. The integrated care plan aims to organise and schedule a set of health and social care services that meet the patient's needs.

The clinician who produces the plan acts on behalf of the patient by collaborating with various health and social authorities and trusts in order to formulate and manage the integrated care plan for the patient. The clinician acts in a 'classical' broker role, providing a 'broker service' based up their expertise, that collects information about the provision of services from service providers and negotiates with them based on the interest of the patient. The integrated care plan is produced once there is agreement between the clinician and providers.

The role of the clinician leads to the research question "How can we support the development and management of an an integrated care plan by using a service broker architecture?" For the case of the IBHIS system, the information broker is only used for the specific purpose of gathering information for the client. However, IBHIS uses a service-oriented architecture which has the potential to be extended to provide more complex forms of 'broker service' to the client or to other brokers.

Thus, the extension of the IBHIS architecture has involved creating a new and additional service broker for care planning, called a care planning broker, where this structure can offer several advantages as follows.

1. Unlike the role of the broker in IBHIS, the care planning broker is not limited to perform only the one specific role of information gathering. It performs an extended role by *interacting* with the care planning services on behalf of the client.

2. The care planning broker not only gathers and blends information from the service providers, as performed by the information broker, but also *negotiates* with the service providers to create the care plan. Both sides can negotiate without knowing details of such things as their location and the format of messages. These details are handled by the information broker, acting as an information mediator, when exchanging information between the care planning broker and the providers.

3. The role of the information broker is extended to *provide* information not only to the client directly but also to the care planning broker, and hence forms an element in a service supply chain. The information broker acts as a primitive service providing information to the care planning broker, which in turn forms the highest layer of the service supply chain.

4. From the perspective of the DAS, the care planning broker also extends the

role of *data access*, which in IBHIS is based on a passive role (read-only), by which access control to data resources is based upon information content and the user's role. The user must be authorised for particular roles and teams, such as the role of a physiotherapist in a multi-professional assessment team, in order to access data services. However, to support the negotiation mechanism between the care planning broker and the DASs, the role of data access now has to become active (read-write). Before accessing data resources, negotiation based on content will take place. If there is an agreement between participants, the content in the data resources will then need to be updated.

5.2 The Nature of the Integrated Care Plan

An integrated care plan, as used within the health care sector, is generally created as a standard clinical procedure. It provides details about the roles and responsibilities of individual health professionals in preparation for an operation, in the immediate post-operative care period, and during rehabilitation and convalescence for the patient. The clinician is the person who has responsibility for managing the integrated care plan by collaborating and negotiating with the service providers, based on the needs of the patient and the constraints of the providers. Re-negotiation of the integrated care plan is also likely to occur as evolving patient requirements may lead to a significant deviation from the original care plan.

This section describes the nature of the integrated care plan. It starts with a review of healthcare in the UK, providing brief information about various kinds of health and social care. Then, it describes the relationship between the case manament and care planning. Finally, it examines how creation of the integrated care plan is related to the idea of a 'classical' broker.

5.2.1 Healthcare in the UK

The Department of Health (DoH) is a department of the United Kingdom government. It aims to "improve the health and wellbeing of people in England" by "providing health and social care policy, guidance and publications for National Health Service (NHS) and social care professionals" (DoH-COI, 2007; DoH, 2010). These various professionals from a wide range of organisations deliver healthcare and social care services to the public as described in the following.

Healthcare

Healthcare provides a broad scope of healthcare services and facilities to people through many kinds of healthcare services, for instance primary care, secondary care, integrated care, emergency care, and healthcare for long term conditions.

1. *Primary care* is the key part of healthcare services. It involves a wide range of services performed by various healthcare professionals, such as general practitioners (GPs), pharmacists, dentists and midwives, who are located in local communities near to the patients. For most needs, the patients have to contact them first and the patients may then be referred to the other health and social services.

2. *Secondary care* provides elective and emergency care through medical specialists in a hospital or other secondary care setting. A primary care professional, such as a GP, normally refers a patient to secondary care, such as day surgery, when the primary care professional diagnoses the patient as needing special treatment from specialist medical care, or surgery in a hospital.

3. *Integrated care* is the cooperation between health and social care providers providing the right treatment and care to the patient.

4. *Emergency care* provides treatment for people, when they need it as a matter of urgency, often after some of accident.

5. Healthcare for *long term conditions* addresses the needs of people who are limited in their daily living from chronic illnesses, for instance heart disease, and chronic obstructive pulmonary disease. This healthcare support enables older people to live independently for as long as possible and involves diversity of health and social care services which help meet both the medical and non-medical needs of people. An example of such healthcare services is continuing care. Continuing care is professional care given to meet the physical or mental health needs of adults with a disability, injury or illness on over an extended period of time.

Social care

Social care provides various types of social care delivered to both children and adults. There are a number of groups of children and adults, such as the elderly and disabled people, including people who have autism and mental problems. These services are provided through a range of social care settings, for instance the individual's home, residential or nursing homes and day centres. The provisions of social care are for all people equally. Moreover they are relied upon as a means of helping people to maintain their independence, control and dignity.

Health and social care services are provided to people through a wide range of organisations, especially the NHS and the social community. The NHS aims to provide a broad scope of health care services through several kinds of NHS Authorities and Trusts located in the local area. The Primary Care Trust (PCT) is a key part of the NHS. It provides a wide range of forms of primary care, such as NHS walk-in centres and the NHS Direct telephone service. These services are managed and run by various professionals, such as doctors, dentists, opticians and chemists. Social care is usually provided through local government agencies.

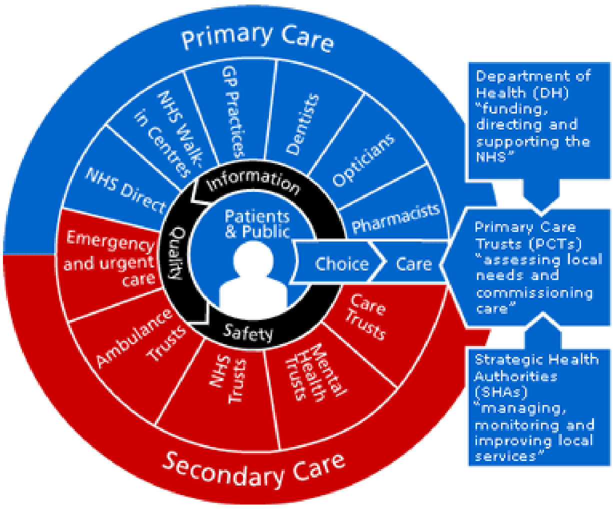

Primary Care
GP Practices
Dentists
Opticians
Pharmacists
NHS Walk-in Centres
NHS Direct
Information
Patients & Public
Choice
Care
Quality
Safety
Emergency and urgent care
Ambulance Trusts
NHS Trusts
Mental Health Trusts
Care Trusts
Secondary Care
Department of Health (DH) "funding, directing and supporting the NHS"
Primary Care Trusts (PCTs) "assessing local needs and commissioning care"
Strategic Health Authorities (SHAs) "managing, monitoring and improving local services"

Figure 5.1: The structure of NHS taken from (NHS, 2010)

Figure 5.1 illustrates the diagram of the NHS structure that is composed from two main types of healthcare: primary and secondary care (NHS, 2010; DoH, 2010).

- *Primary care*

  Most patients first contact the PCT and the primary care is delivered by various independent contractors, involving GPs, dentists, pharmacists and optometrists.

- *Secondary care*

  Secondary care addresses acute healthcare. There are two types of such care: elective care and emergency care.

  - Elective care implies medical care or surgery that is planned and treated by specialists.
  - Emergency care is the form of healthcare service provided to people who need medical advice, diagnosis and/or treatment responsively and urgently. Accordingly, a range of different types of service exist, from ambulance services and A&E departments to advice from NHS Direct

or GPs. A patient is usually referred to a specialist or professional group for elective care.

5.2.2 Case Management and Care Planning

“Case management is a service led by a community matron or case manager that provides proactive, coordinated care to people who have a complex mix of health and social care needs. It provides an intense level of care, preventing people from unnecessary admission to hospital and providing care in the person’s home or community setting” (DoH, 2010). The intense level of care consists of professional long-term care provided to support the health, psychological, social and rehabilitative needs of a patient. Case management aims to provide co-ordinating services or care packages according to the needs of an individual person as assessed by a range of health care professionals (Neies and Berman, 2004).

The case manager has responsibility for integrating the various health and social care professionals and systems working together to provide coordinated care to people. The professionals from different aspects of health and social care work together as multidisciplinary teams, suitably mixing their skills to meet people’s needs for coordinated and seamless care. The challenge for service provision is to balance needs and resources under deficiency and option.

There are different types of approaches or models to case management, for instance the intensive case management model and the independent brokerage model. The first model is for a patient with severe and complex needs, whereas the brokerage model is like service brokrage. An independent agency employs the case manager to tailor services by working with the patient and a carer to define their needs, supply information and assist them to access appropriate services.

Case management consists of five main stages described as follows (DoH, 2010; Neies & Berman, 2004; Rose, 1992):

1. *Case-finding and screening*

   A professional assessment team initially identifies the comprehensive needs of a patient and assesses whether the patient is eligible for the provision of a service. If eligible, the patient is referred to the integrated health care team for needs assessment. The health care professionals are the people in the health and social care authorities and trusts who deliver the health and social care services to the patient in a systematic way.

2. *Needs assessment*

   The integrated health care team makes an integrated assessment of the patient's comprehensive needs and problems as well as the present and potential strengths and weaknesses of the patient. The patient's needs may take into account such circumstances as, for instance, health, personality, social context, economy, education, mental health, ethnic and cultural background. The information from the needs assessment is routed to the clinician to assist with planning an integrated care plan.

3. *Planning and delivery of care*

   The clinician develops an integrated care plan that meets the patient's needs and problems as determined from the process of needs assessment. Then the resulting package of the health and social care services is implemented and delivered to the patient by several health and social care professionals depending on the context of their work. Practices and specialist nurses deliver the health care services in GP practices and hospitals respectively, while social care support is provided people in their homes by such roles as community matrons and social care workers.

4. *Monitoring and review*

The clinician or case manager will periodically contact the patient and the integrated care team to make sure that the services planned are being provided and to assess their effect. The healthcare professional team regularly visits the patient and reassesses the adequacy and appropriateness of the health and social care services provided. The clinician receives the feedback on progress from the healthcare professional team. If the feedback indicates a need to change the integrated care plan, the integrated care team and the clinician work together to rearrange this.

5. *Closure*

   The patient's case may be closed because of many reasons as in the following examples.

   - The objectives of the integrated care plan for the patient are achieved.
   - The integrated care team withdraws.
   - The patient quits the care services because of his/her own reasons.
   - The regular service sector and the family carers take over care of the patient.

5.2.3 The Integrated Care Plan and a Care Planning Broker

The integrated care plan is a care plan for an individual patient with complex health and social care needs (Rigby, 2005; Neies and Berman, 2004). It is a standard clinical procedure, that provides details about the roles and responsibilities of health professionals in preparation for hospital and social care treatments. One example is that of a surgical operation, where the plan describes how the immediate and post-operative care as well as rehabilitation and convalescence are to be delivered.

A clinician or a case manager represents the role of the patient, collaborating with a number of the health and social authorities and trusts to produce and manage the integrated care plan. The clinician finds and chooses the appropriate health care service providers and manages the case to provide the suitable services or treatments for the patient.

However, providing appropriate quality of care to satisfy the patient's needs may need negotiation between the case manager and the service providers to create the package of services for the integrated care plan. This negotiation deals with the package procedures concerning patient needs and any preferences which deviate from the package. Service negotiation is the process of interacting between the case manager on behalf of the patient and the providers with the aim of agreeing the terms of any condition relating to supplying the service. In addition, the aims of negotiation in integrated care planning is not only to satisfy patients but also to satisfy other conditions such as reducing risk, and achieving an acceptable price for the services.

The role of the clinician is that of a 'classical' broker with regard to the management of the integrated care plan. The following list reflects three main characteristics of the concept of the broker.

- The clinician performs tasks on behalf of the patient as the end-user of the service. The clinician is as a trusted person who is given authorisation by the patient to deal with the service providers, based on the interest of the patient and their own expertise.

- The clinician contacts the service providers and collects information about their service provision from them. Then the clinician blends it and puts all neccessary information into the integrated care plan for the patient.

- The clinician negotiates with a number of the service providers based on the needs, preferences and conditions of the patient. If there is agreement between them, then the integrated care plan will be created or modified for the patient.

Therefore, the case management or care planning of the integrated care plan is an example of a service that can be provided by a broker. This can compare with the other classical example of this role, a stockbroker who is a professional broker acting on behalf of investors buying and selling shares as well as other securities through market makers. In this case, the client asks the stockbroker to make a transaction for trading in the stock market for him/her.

5.3 The Care Planning System

The conceptual framework of the care planning system is based on the concept of developing the integrated care plan via a 'service broker' that provides a service to create and manage the integrated care plan for the patient. This section starts with the definition and structure of the integrated care plan for the purpose of this investigation. Then, the conceptual framework of the care planning system is outlined in terms of its components. After that, we describe how the system performs to produce and manage the integrated care plan.

5.3.1 The Integrated Care Plan

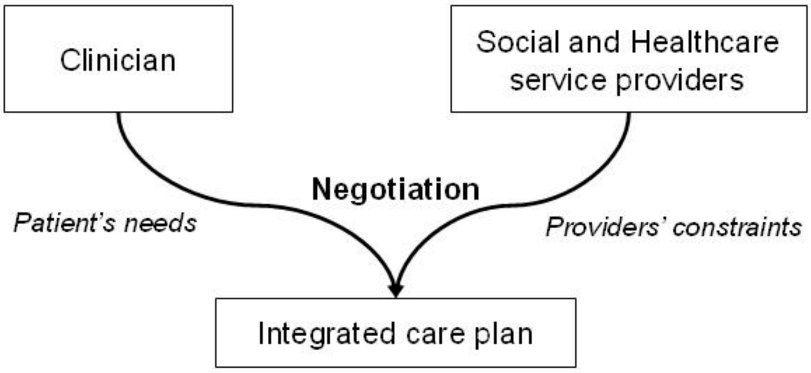


Figure 5.2: The negotiation framework for creating the integrated care plan

Figure 5.2 shows the negotiation framework used for creating the integrated care plan. The clinician acting on behalf of a patient contacts and negotiates with various types of social and health care service providers. This negotiation is based on the patient's needs and preferences and the service providers' constraints. The result of the negotiation should be an integrated care plan meeting the needs and preferences of the patient, and agreements for the term and conditions of services provided to the patient by the providers.

The integrated care plan or the care plan contains a set of services for a given form of health and social care episode that are needed by the patient. Each service specifies the needs and preferences as well as the constraints of the patient, such as date, time, cost and precedence among related services. With the agreement from the patient, the clinician can change the details of the care plan and negotiate with the service providers for the provision of their services, when the patient's conditions are changed.

5.3.2 The Conceptual Framework

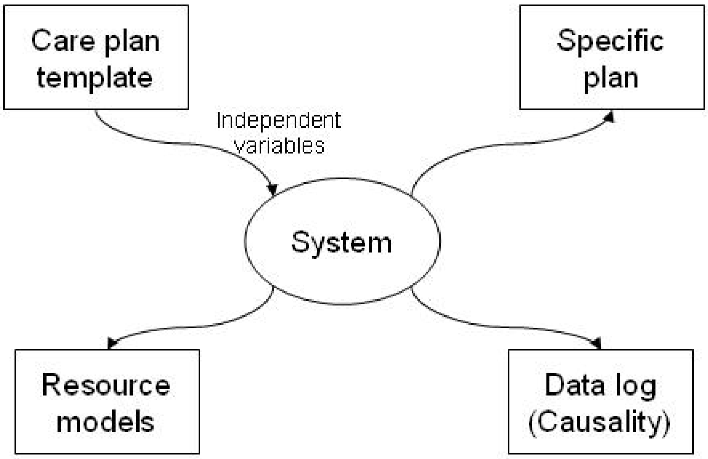


Figure 5.3: The conceptual framework

Figure 5.3 shows the conceptual framework of the system as used to create a specific integrated care plan to meet the patient's needs, such as the integrated care plan for patients with a fractured neck or femur. The service negotiation occuring within the system is between a planning broker and the service providers. The system involves four main components: care plan tempate, specific plan, resource models and data log.

Care plan template

The care plan template consists of a template that groups elements of the integrated care plan. Each care plan template is defined in terms of the set of services needed for a given form of health and social care episode. For each service, it specifies the needs and preferences as well as the constraints of the patient, such as date, time, cost and precedence among related services.

The clinician chooses the care plan template suited for the patient. Then the clinician modifies the needs, preferences and constraints in the template so that they are appropriate for the patient. The modified template is finally submitted to the system to create a specific plan for the patient.

Specific plan

The specific plan is the result of the process of service negotiation that occurs between the planning broker and the relevant service providers. The specific plan of integrated care addresses the needs of an individual patient. It identifies the provision of services by various service providers, based on the terms and conditions agreed between the care planner, the patient, and the service providers. There are two types of specific plan: the care plan and the actual care plan.

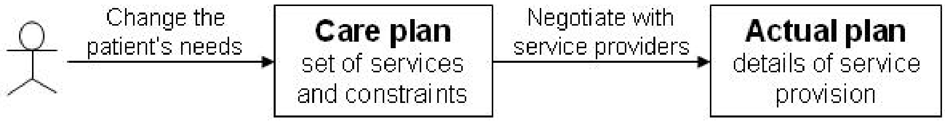

Change the patient's needs
Care plan
set of services and constraints
Negotiate with service providers
Actual plan
details of service provision

Figure 5.4: The relationship between care and actual plan

1. *Care plan*

   The care plan provides details about a set of health and social care services together with their pre-defined needs and constraints. It uses the same structure as the care plan template. When the patient's needs or conditions are changed, the clinician changes the details of the care plan and uses this to negotiate with the service providers for the provision of their services. The result of the negotiation is the actual care plan.

2. *Actual care plan*

   The actual care plan holds the details about the service provision of integrated care for a given patient and episode of care. It mainly contains details about date and time for each service that is to be provided to the patient by various service providers and of the provider. The actual care plan can also be changed should the patient's condition change.

Resource models

The resource models hold the data about any resources that are related to the integrated care plan. For the purpose of this research project, the resources for the individual service providers are represented by their diaries, specifying the available dates and times for providing their resources.

Data log

The data log contains information about system actions and is used to examine how well the system's components perform according to expectation. The information is recorded during service negotiation between the planning broker and the service providers.

5.3.3 System Operation

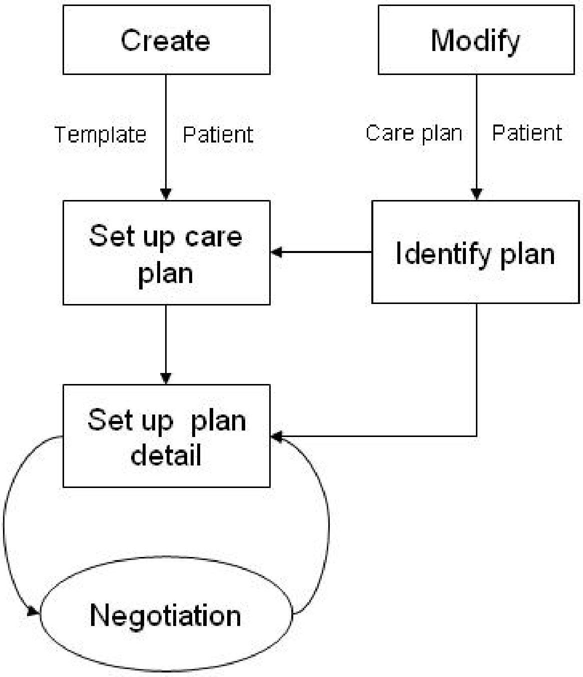


Figure 5.5: System operation for the integrated care plan

Figure 5.5 presents the system operation for creating the integrated care plan. There are two main operations involved: creating and modifying of the integrated care plan. Both of them share the same operations for setting up the care plan and its actual care plan. The create operation is used to create a new care plan and its actual care plan from a care plan template that is suited for the patient. Then the modify operation is employed to modify the new care plan or its actual plan when they are needed to be changed because of the patient's changing conditions. The following explains more detail for both operations.

Create

1. Retrieve and choose a care plan template that is suited for the patient
2. Change the details of the template according to the needs, preferences and constraints of the patient
3. Issue a set of requests to the providers
4. Negotiate with the providers

   The providers make reservation in their diaries for resources that are available to provide services on dates and times requested by the patient
5. Select the providers' offers that meet the needs of the patient
6. Accept the selected offers
7. Acknowledge the results of offer selection to the providers

   The providers update the status of the reserved resources in their diaries by performing the following operations.

   - Book the selected offers
   - Decline the offers that were not selected
8. Create a new care plan and its actual plan for the integrated care

Modify

1. Retrieve a care plan to be modified

2. Change the details of the care plan according to the needs, preferences and constraints of the patient

3. Perform the same actions as for the 'Create' operation, from from steps 3 to 7

4. Update the care plan and its actual plan

5.4 An Example Care Plan

This section presents an example care plan, illustrating how the care planning system is used to create the integrated care plan, and demonstrating this with an example use case and two scenarios derived from this.

5.4.1 Use Case

> Mrs Edith Arbuthnot is a 72 year old widow, living alone in the family home in a large village some 15 miles west of the market town. Her husband having been a self-employed general builder, she has no mortgage but equally few savings, and her sole formal income is her State pension. She has two married children – a daughter Penny with school-age children, who lives 30 miles away to the east and works in a shop on Monday - Wednesday and Friday – Saturday. Her son Douglas is a veterinary surgeon in private practice, who lives 50 miles away to the north-west and has a young family. Penny often visits on a Thursday afternoon, while Douglas visits about once a month with the grandchildren.
>
> When out shopping, Edith falls and is clearly in pain. An ambulance is called, she is taken to the A & E department, and a fractured neck of

> femur is diagnosed. This is treated by insertion of a pin, and the prognosis is good but with reduced mobility anticipated (initially needing a walking frame, but with expectation that she will be able to manage with a walking stick). As soon as her hip is stable, but while the wound is still being dressed, Edith is moved to an Elderly Rehabilitation ward. Discharge within 10 days seems possible on medical grounds, and the discharge liaison nurse starts to plan her possible discharge.
>
> (Rigby, 2008)

5.4.2 Scenarios

The consultant consults and assesses Edith for her needs and constraints. Then the consultant passes the result of Edith's assessment to the nurse to create the integrated care plan for her treatment and support after discharging from the rehabilitation ward, as descibed in the first 'Create a Care Plan' scenario. However, when her conditions deviate from the package of care, the nurse modifies her care plan and renegotiates with the providers for the new required or canceled services as shown in the second 'Modify the Care Plan'. The two cases shown in the following scenarios are adapted from (Rigby, 2008).

5.4.2.1 Scenario: Create a Care Plan

Case

> Edith will need physiotherapy for her injury for four weeks. This treatment can be done at a community hospital chosen by her from a number provided by a NHS Trust that is near her home. She prefers the treatment should be held on Tuesday and Thursday afternoons. Nevertheless, she will have physiotherapy after a physiotherapy assessment by the senior physiotherapist

there. Sitting case ambulance transport will also be needed for the physiotherapy. This kind of transport is contracted by the PCT to a several transport providers. The availability of the assessment, the physiotherapy and ambulance transport has to be established.

Operation

The nurse uses the care planning system to create the integrated care plan for Edith as in the following.

1. Retrieve and choose a care plan template that is suited for Edith's case
2. Change the details of the template by identifying the needs, preferences and constraints of Edith
3. Issue the requests to the system
4. Select the providers' offers that meet Edith's needs
5. Accept and save the new care plan to the system's database

5.4.2.2 Scenario: Modify the Care Plan

Case

The senior physiotherapist assesses Edith after three weeks of physiotherapy treatment and finds that she does not need the treatment any more. Therefore, the assessment, the physiotherapy and ambulance transport have to be cancelled.

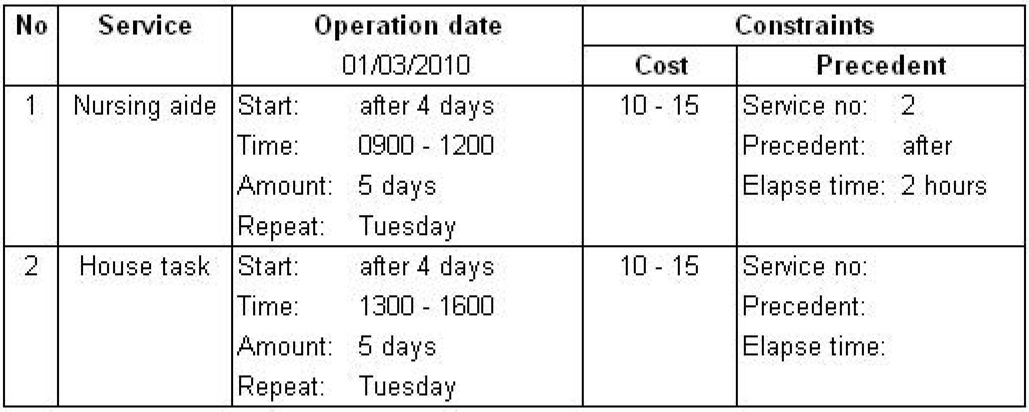
However, she needs assistance to cope with daily living at home, due to her reduced mobility. Penny cannot come to assist her with bathing for a few weeks, which she normally does once a week. Hence, the nurse has to plan for a nursing aide from community nursing services to undertake this until Penny comes back to do the bathing for Edith by herself.

Operation

The nurse uses the care planning system to modify the integrated care plan for Edith as in the following.

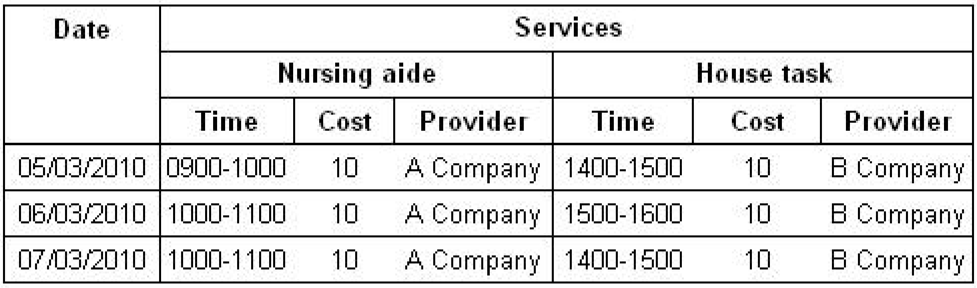
1. Retrieve Edith's care plan from the system's database
2. Change the details of the care plan by canceling the assessment, the physiotherapy and ambulance transport, and then adding the requirement for a nursing aide to the care plan.
3. Issue the requests to the system
4. Select the providers' offers that meet Edith's needs
5. Accept and update the care plan to the system's database

Tables 5.1 and 5.2 illustrate the example of a care plan request and the actual plan respectively after the 'Modify' operation.



No	Service	Operation date	Constraints	
		01/03/2010	Cost	Precedent
1	Nursing aide	Start: after 4 days Time: 0900 - 1200 Amount: 5 days Repeat: Tuesday	10 - 15	Service no: 2 Precedent: after Elapse time: 2 hours
2	House task	Start: after 4 days Time: 1300 - 1600 Amount: 5 days Repeat: Tuesday	10 - 15	Service no: Precedent: Elapse time:

Table 5.1: An example of a care plan request



Date	Services					
	Nursing aide			House task		
	Time	Cost	Provider	Time	Cost	Provider
05/03/2010	0900-1000	10	A Company	1400-1500	10	B Company
06/03/2010	1000-1100	10	A Company	1500-1600	10	B Company
07/03/2010	1000-1100	10	A Company	1400-1500	10	B Company

Table 5.2: An example of an actual plan

5.5 Summary

Chapter 5 has examined the nature of the integrated care plan that forms the basis of the research case study. An integrated care plan is commonly created as a standard clinical procedure. It provides details about the roles and responsibilities of health professionals in preparation for hospital and social care treatements. A clinician uses and manages the integrated care plan by collaborating and negotiating with providers, based on the needs of a patient and the constraints of the providers.

The conceptual framework of the care planning system is used to create an integrated care plan to meet the patient's needs. The system consists of four main components: care plan template, specific plan, resource models and data log. The care plan template is defined the set of services needed for a given form of health and social care episode. The specific plan is the result of the process of service negotiation that occurs between the planning broker and the relevant providers. The resource models hold the data about any resources related to the integrated care plan. The data log is used to record information during negotiation between negotiation participants to examine the performance of the system's components.

There are two main operations of the care planning system. The create operation is used to create a new care plan for the client: create and modify. Then the modify operation is used to modify the new care plan when they are needed to be changed because of the patient's changing conditions. The care planning system is demonstrated by using a use case and its scenarios.

Chapter 6

The CAPTAIN Broker

Chapter 5 explored how the concept of the broker as well as service negotiation could be extended to involve the development of a care plan. This chapter describes an implementation of the broker concept to demonstrate those ideas.

6.1 Introduction

IBHIS is an information broker that is based on the concept of SaaS. It performs by providing services that deliver integrated data that has been gathered from distributed, heterogeneous and autonoumous data sources. There is potential to extend it for various aspects of service-based software, such as data access control (Budgen *et al.*, 2007), as well as to extend the concept of the broker (Budgen *et al.*, 2005). The information broker itself provides a service by acting as the mediator between the client and the service providers. Consequently, three main features of the IBHIS broker and SaaS provide scope for extension of the concepts involved in IBHIS: the 'broker service', the nature of the service supply chain and service negotiation.

1. The 'broker service'

   The main purpose of the information broker as realised through IBHIS is to provide information service helping Health and Social clinicians and professionals in their decision-making activities (Budgen et al., 2004). The information broker dynamically locates and gathers information ‘on the fly’ from a number of healthcare service providers and then supplies it to fit the needs of the client. However, the client may also need other forms of the ‘broker service’, for instance a health insurance broker service. A health insurance broker should be an independent agency providing services to help the client find health coverage from several health insurers in order to meet the client’s needs and conditions. Therefore, there is potential to extend the role of the broker by providing a number of various ‘broker services’ to the client, for instance a health insurance broker service. The health insurance service broker should be an independent agency providing services to help a client find health coverage from several health insurers to order to meet the client’s interest and conditions. Therefore, there is potential to extend the role of the broker by providing a number of various ‘broker services’ to the client.

2. The nature of the service supply chain

   “Services are supplied through a service supply chain” (Bennett *et al.*, 2000). Thus the client can receive a service that is itself composed from a number of services. The information broker is initially intended to supply the information to the client. It can also form a primitive service in a hierarchy of service providers by providing its information to the other service brokers, through a consumer-supplier (retail) market. The service brokers in the higher layers of the hierarchy of service providers do not have to deal with the original data sources directly and can focus on their roles and let the information broker deal with the distributed, heterogeneous and autonomous data sources. This implies that the information broker can act as an information mediator between the other types of the broker services and the data services. From the perspective of the broker service, the service broker itself can provide its services by composing services from a group of various broker services to achieve a specific task for the client.

3. Service negotiation in IBHIS

   The IBHIS application employs service negotiation for end-user authorisation to access data services (Budgen *et al.*, 2004). It does not deal with any conflict of interests between the end users and service providers to meet a mutually acceptable agreement on such issues, as cost. In particular, the information broker provides services that exhibit 'read-only' properties in terms of negotiation. The service negotiation performed on the specific domain of application only involves 'read-only' access to healthcare data. Consequently, the role of service negotiation function can potentially be extended to address a full range of service characteristics.

To investigate these issues, especially the service negotiation in IBHIS, we have developed a new system, called CAPTAIN (Care Planning Through Auction-based Information Negotiation). The purpose of the CAPTAIN system is to create integrated care plans within the healthcare sector to provide a research case study.

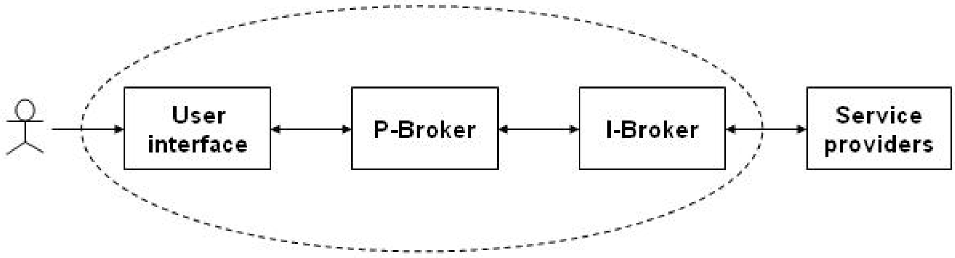


Figure 6.1: The structure of CAPTAIN

Figure 6.1 shows the diagram of the structure of CAPTAIN. It consists of two main service brokers: the care planning broker or P-Broker; and the information broker or I-Broker. The P-Broker provides care planning services by producing the integrated care plan for the client. It negotiates with several health care service providers based on the needs of the client. During service negotiation, the I-Broker performs as an intermediary by identifying relevant providers of services, and organising exchange of the messages between the P-Broker and the service providers.

The system of CAPTAIN has been adapted from the architecture of IBHIS by extending the concepts and role of the service broker used in IBHIS. According to the concept of the service supply chain, CAPTAIN forms a set of services for the client by including the P-Broker to combine with the I-Broker. The two service brokers perform different roles to provide different services to the client or the other service brokers. The I-Broker provides an information service that is a primitive service in a hierarchy of service providers, providing information to the P-Broker. The P-Broker does not have to deal with the original data sources of the service providers directly and can focus on the care planning services and let the I-Broker deal with the providers that their data sources are distributed, heterogeneous and autonomous. In addition, CAPTAIN can extend new professional service brokers and combine them with the set of service brokers to form a new service supply chain for the client.

The following are the main activities used for developing and building CAPTAIN's concept and its system, as well as for its testing and evaluation.

1. Develop the system architecture of CAPTAIN

   The architecture of CAPTAIN was developed by combining the P-Broker and the I-Broker into CAPTAIN to provide care planning services for the client.

2. Develop a negotiation model for CAPTAIN.

   The selected negotiation models from the various contexts of negotiation research have been used to develop a negotiation model for CAPTAIN by adapting those concepts and elements that are suited for the aims of CAPTAIN.

3. Define the key issues and problems of the integrated care plan

   The key issues and problems in healthcare are addressed in the integrated care plan. Then the negotiation model is deployed into the integrated care plan.

4. Incorporate negotiation elements into CAPTAIN

   The CAPTAIN application has been designed and built based using the architectural model from the IBHIS application. Elements of negotiation have been incorporated into the CAPTAIN application.

5. Test and evaluate the CAPTAIN application

   The CAPTAIN application has been built using Java 2 Enterprise Edition (J2EE) Web Services technologies. The incorporated negotiation elements and the care planning broker for CAPTAIN have been tested and evaluated according to the criteria defined by these.

The P-Broker of CAPTAIN should perform the role of a care planning broker that gathers the needs and preferences of the client. Then it negotiates with the service providers via the I-Broker to produce the integrated care plan for the client based on the client's interest.

6.2 The Architecture of CAPTAIN

Figure 6.2 illustrates the architecture used for CAPTAIN. It extends the broker form used in IBHIS (Budgen et al., 2004) to accomplish the objectives of this research. Its overall architecture consists of four main service components: the P-Broker, the I-Broker, DASs, and the semantic registry. The following describes more details about the service components of CAPTAIN.

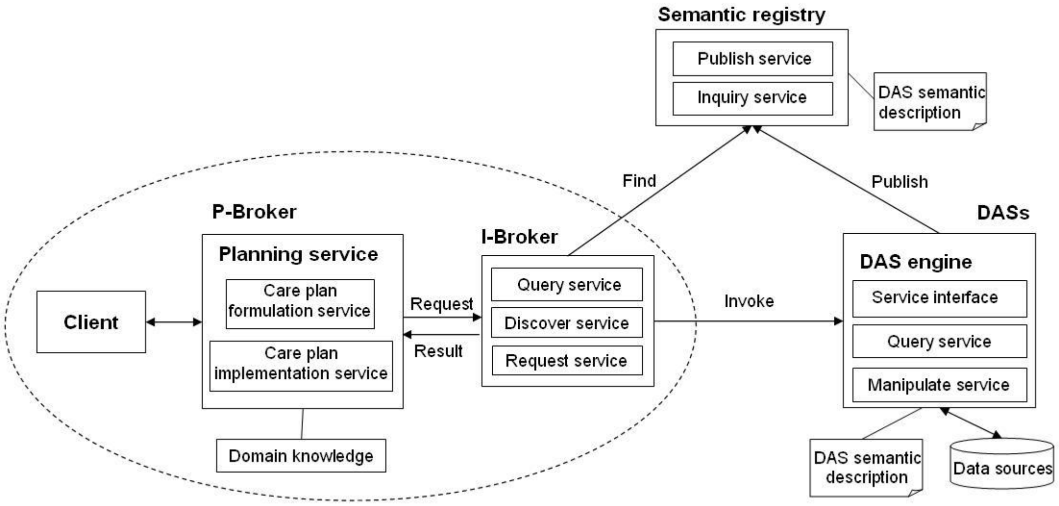


Figure 6.2: The architecture of CAPTAIN

DASs

The DASs are a set of heterogeneous, autonomous and distributed data sources within the healthcare sector. They consist of several kinds of health care service providers, providing information about health and social care resources to the I-Broker. This information is related to the needs of the integrated care plan, such as appointments, health care professionals and the service description of the DASs. The DAS engine of each DAS consists of “Service interface” and two main service functions: “Query service” and “Manipulate service”.

- The “Service interface” defines a set of data service types for the I-Broker.
- The “Query service” handles queries from the I-Broker. It retrieves data from the data sources according to the I-Broker’s queries.
- The “Manipulate service” deals with the manipulation of the agency’s data sources according to the I-Broker’s queries.

I-Broker

The I-Broker acts as a information mediator between the P-Broker and the DASs. Both of them exchange messages via the I-Broker so that they do not have to know detailed information about each other, such as message format and locations. The I-Broker takes action in the interests of the P-Broker. It uses requests from the P-Broker to collect information from the DASs. Then, it returns the result to the P-Broker by blending the information from the DASs into the composite form defined by the P-Broker. The I-Broker consists of three service functions: "Discover service", "Query service" and "Request service".

- The "Discover service" is used by the "Query service" and the "Request service" to locate the DASs and to invoke their data services for the P-Broker.
- The "Query service" deals with the requests from the P-Broker. It formulates queries from the P-Broker's requests and issues the queries to request offers from the DASs.
- The "Request service" handles the P-Broker's request by issuing the requests, such as 'accept', 'reserve' and 'decline' offers to the DASs.

P-Broker

The P-Broker is a new "service broker" that extends the IBHIS architecture. For CAPTAIN, the P-Broker provides a care planning service to produce the integrated care plan for the client. It uses data and information provided by the DASs via the I-Broker, together with by its planning knowledge repository called "domain knowledge". The P-Broker contains a "Planning service" component which consists of two main service components: "Care plan formulation service" and "Care plan implementation service".

- The "Care plan formulation service" is used to analyse a client's request and formulate the requests or queries for the I-Broker.

- The "Care plan implementation service", then uses the requests from the "Care plan formulation service" to collect and analyse the information from the DASs, via the I-Broker, in order to create the integrated care plan for the client.

Semantic Registry

The semantic registry holds information about the DASs' service descriptions. There are two main service components: "Publish service" and "Inquiry service". Both service components are used by the DASs and the I-Broker to publish and find the DASs' service description respectively.

To achieve the aims of this research, CAPTAIN has extended the concept of the DAS of IBHIS in two main features: data provision and the role of data access.

1. The data provision of IBHIS focuses on patient-centric issues. However, CAPTAIN has to deal with the service negotiation between the P-Broker and the DASs that is based on health and social care resources. Therefore the data provision of DASs in CAPTAIN involves service-centric issues that involve data and information that is related to negotiation issues for creating the integrated care plan.

2. For the role of data access, the DASs of IBHIS perform in a passive role (read-only). The DAS engine accesses data sources to query content on a read-only basis. The resource access control is based upon information content and the user's role. These features have been extended for CAPTAIN to provide an active role (read-write). Before accessing the data resources, negotiation ba-

sed on content will take place. If there is an agreement between participants, the content in the data resources will be updated.

For IBHIS, the service description of the DASs includes many kinds of information, for instance the specification of input and output data as well as its format; and the access rules for using the data (Budgen *et al.*, 2005; Raiffa, 1982). However, the information broker of CAPTAIN does not need to deal with these issues. It only uses simple data, for instance a service name, to discover the service descriptions of the DASs. Therefore the semantic registry of CAPTAIN retains the basic information about their service descriptions, such as DAS's name, service name and its service access address.

6.3 Negotiation Model

Service negotiation in CAPTAIN is performed between the P-Broker and the DASs via the I-Broker. The P-Broker and the DASs act as service requestor and service providers respectively. The approach used for this negotaition uses an auction model and is based on the conflict of interests between the service requestor and the service providers. The negotiation is one-to-many or multi-bilateral negotiation protocol in which the P-Broker concurrently negotiates with several DASs. Both sets of participants negotiate over multiple independent attributes or issues according to the client's request. In addition, the aim of their negotiation interaction is focused on the client's needs or preferences. Accordingly, the DASs generate and propose their offers, within their constraints, in order to satisfy the client.

To deal with such a negotiation situation, the negotiation model developed for CAPTAIN is based on several automated negotiation models from various aspects of negotiation situations; especially Web services (Hung *et al.*, 2004), autonomous agents (Jennings *et al.*, 2001), service-oriented software (Elfatatry and Layzell, 2004), and E-business (Kim & Segev, 2003; Li *et al.*, 1998). There are three common negotiation elements used in these: negotiation object, negotiation protocol and decision model.

- A *negotiation object* or message is a group of issues exchanged and negotiated by negotiation participants to reach their agreement.
- A *negotiation protocol* identifies the rules of actions or a mechanism followed by negotiation participants during the negotiation.
- Each participant also has its own *decision model*. This is a decision-making process used to make an offer or counter-offer in response to that from the other participants. Each participant may apply a different policy, strategy and tactic for the decision model.

The elements of negotiation for CAPTAIN have been chosen based on two main features: the nature of the integrated care plan and the conventional negotiation model. For the first feature, there are three fundamental stages involving in producing the integrated care plan: before, during and after negotiation.

1. In the planning stage: the clinician prepares initial information for service negotiation by gathering information from a patient, choosing an appropriate care plan template and modifying the care plan to suit the needs of the patient.
2. During the negotiation stage: the clinician uses the broker to find the providers and negotiates with them to produce the care plan for the patient.
3. After the negotiation stage, the clinician chooses between the offers, based on the patient's interest, and then accepts or declines the different providers' offers.

These fundamental stages led to the idea of including the negotiation process element into the negotiation model for CAPTAIN. The negotiation process consists of three basic stages: pre-negotiation, service negotiation and post-negotiation.

For the second feature, service negotiation needs to use a conventional negotiation model that consists of three basic negotiation elements: object, protocol and decision model. Most of the negotiation models available include these negotiation elements in their negotiation model. So the negotiation model for CAPTAIN includes the negotiation object, negotiation protocol and decision model.

The negotiation model for CAPTAIN combines the negotiation process, negotiation object, negotiation protocol and decision model needed to deal with the negotiation situation that arise when creating a care plan. The negotiation model for CAPTAIN is shown in figure 6.3.

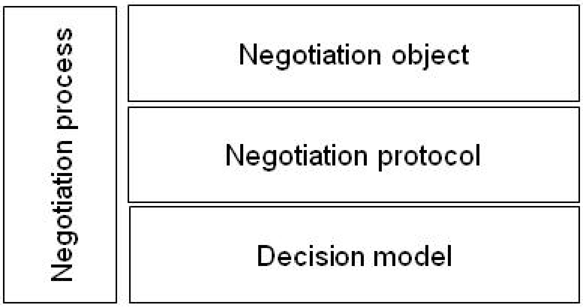


Figure 6.3: The negotiation model of CAPTAIN

6.3.1 The Negotiation Process

The negotiation process is a dynamic end-to-end process starting from the user request and ending up with a resolution. CAPTAIN defines three core sequence stages of negotiation process: pre-negotiation, service negotiation, and post-negotiation. The negotiation process is explained more fully in Section 6.4.

1. The *pre-negotiation stage* is concerned with producing the initial information for the service negotiation stage. There are three main sequence operations involved in this stage: the publishing of service descriptions by the DASs, the request generation by the client and the query formulation by the P-Broker.

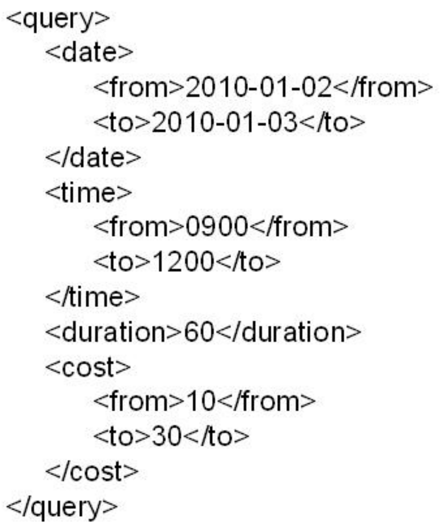
2. The *service negotiation stage* is the stage in which the P-Broker negotiates with the DASs in order to produce the integrated care plan for the client. During the service negotiation process, both negotiation participants exchange their information via the I-Broker.

3. The *post-negotiation stage* deals with the negotiation results from the service negotiation stage. The client is involved in the decision making since the client accepts or declines the results. The DASs will be sent an acknowledgment of the result of the client's decision making. Then the content in their internal data resources can be updated.

6.3.2 The Negotiation Object

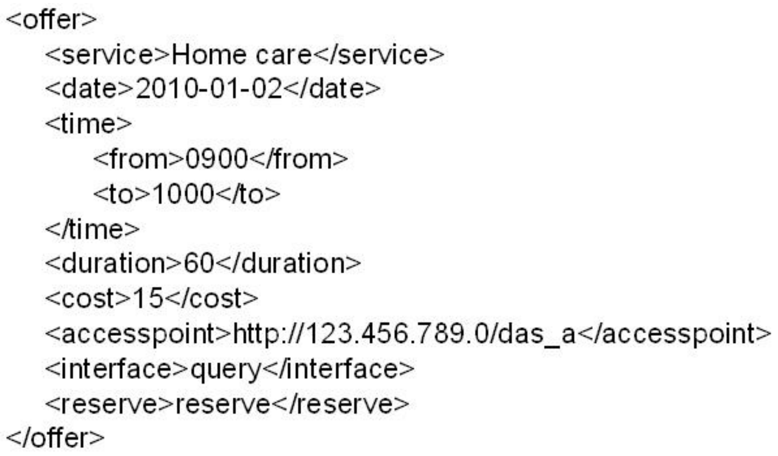
The negotiation object contains the information needed for negotiation between the P-Broker and the DASs in order to produce the integrated care plan for the client. The P-Broker exchanges the negotiation object with the DASs via I-Broker during the service negotiation stage. The I-Broker may extract some information from the P-Broker to find the service description of the DASs from the semantic registry. Then the I-Broker can locate and invoke the data services of the DASs. In addition, the I-Broker may add some information into the negotiation object that will used by the P-Broker and the DASs.

For CAPTAIN, there are three main types of negotiation object: query, offer and acknowledgement. The query is created by the P-Broker proposing the query to the DASs. Then the DASs respond to the P-Broker's query with the offers. Finally, the planning broker sends the acknowledgement of its decision-making according to the offers to the DASs. Figures 6.4 and 6.5 show the example of negotiation objects for a query and an offer respectively.



```
<query>
   <date>
      <from>2010-01-02</from>
      <to>2010-01-03</to>
   </date>
   <time>
      <from>0900</from>
      <to>1200</to>
   </time>
   <duration>60</duration>
   <cost>
      <from>10</from>
      <to>30</to>
   </cost>
</query>
```

Figure 6.4: Example of a negotiation object for a query



```
<offer>
   <service>Home care</service>
   <date>2010-01-02</date>
   <time>
      <from>0900</from>
      <to>1000</to>
   </time>
   <duration>60</duration>
   <cost>15</cost>
   <accesspoint>http://123.456.789.0/das_a</accesspoint>
   <interface>query</interface>
   <reserve>reserve</reserve>
</offer>
```

Figure 6.5: Example of a negotiation object for an offer

6.3.3 The Negotiation Protocol

The negotiation protocol defines the rules and the states of negotiation interaction that all system components of CAPTAIN have to follow during service negotiation process. The aim is for the P-Broker and the DASs to reach an agreement between them. The approach used for service negotiation in CAPTAIN is based on a multi-attribute reverse auction mechanism, whereby the P-Broker, acting on behalf of the client, solicits offers about services from a set of potential DASs.

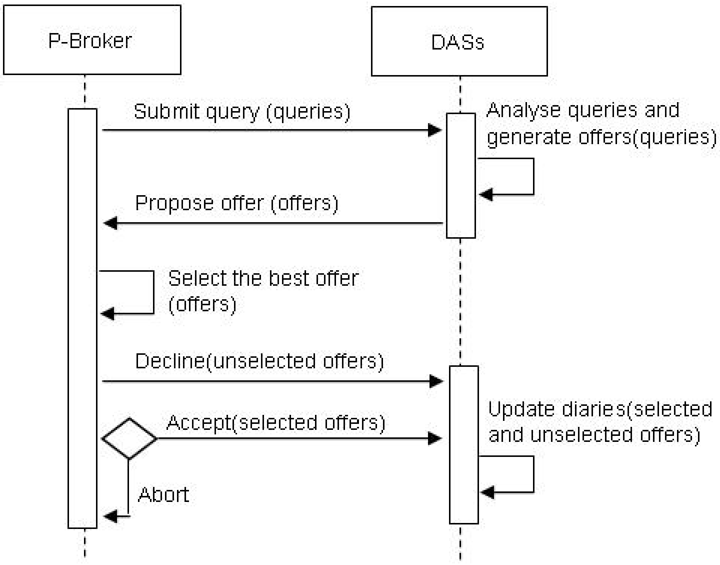

P-Broker
DASs
Submit query (queries)
Analyse queries and generate offers(queries)
Propose offer (offers)
Select the best offer (offers)
Decline(unselected offers)
Accept(selected offers)
Update diaries(selected and unselected offers)
Abort

Figure 6.6: The negotiation protocol of CAPTAIN

Figure 6.6 shows the negotiation protocol of CAPTAIN as described in the following mechanism.

1. The P-Broker submits queries to the DASs.
2. The DASs process the queries as follows.
    (a) Analyse the queries, generate offers for the the DASs' resources that are available, provisionally reserve these for the DASs' service provision for the client
    (b) Propose the offers to the P-Broker
3. The P-Broker processes the DASs' offers as follows.
    (a) Evaluate the offers and select the set of the offers which best satisfy the client's needs and preferences
    (b) Acknowledge the reservation status to the DASs
        i. The decline status for the unselected offers

ii. The accept status for the selected offers

4. The DASs update the reservation status of their' resources for both accepted and unselected offers.

6.3.4 The Decision Model

The decision model consists of the rules employed by the P-Broker and the DASs for decision making. Both the P-Broker and the DASs have their own decision models which are internal and private decision processes. Each negotiation participant has its own strategy which is used to define the plan of decisions or actions needed to achieve its negotiation objective.

6.3.4.1 The P-Broker

The P-Broker uses the information from the client's request to generate queries for the DASs. When it receives a number of the offers from the DASs, it uses its own decision model to evaluate them and to select the offers that are met by the client's needs, preferences and constraints.

However, from the front-end point of view, the client can also define its own strategy in the request. The request consists of four main parts: a service name, date details, constraints and strategies ('earliest time', 'latest time', 'cheapest cost', and 'I decide'). The P-Broker applies the client's strategy in its decision model to evaluate and in order to select the offers preferred by the client.

6.3.4.2 The DASs

When the DASs receive queries from the I-Broker, they use the decision model to generate offers for the P-Broker in response to its queries. Within their decision models, there are three criteria that are of concern: cost, reservation status and timeout.

The DASs generate an offer for the client if the client's requests are acceptable based on these criteria. For the cost, the client's request will be considered if it is in the range of service price specified by the DAS. The other two criteria will be explained as in the following.

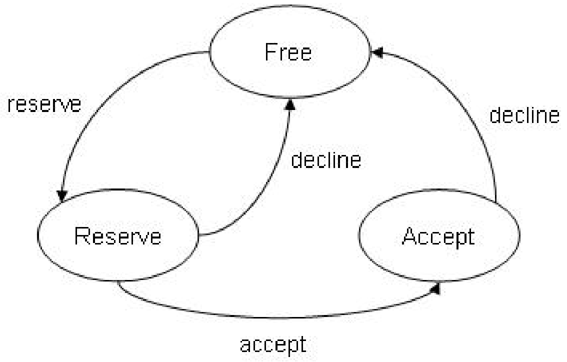


Figure 6.7: The reservation states of a DASs' diary

Figure 6.7 shows the reservation states of DASs' diary. There are three major states used to describe available resources of the DASs: 'free', 'reserve' and 'accept'. The initial state of the diary is 'free'. When there is a service request from a client, the DAS Engine will check the status of reservation in the diary. If the dates and times requested are free, then the relevant diary 'slot' will be reserved. After that, if the client wants to decline or accept the 'reserved' dates and times, the diary will be updated, changing the status of the reservation to 'free' or 'accept' as appropriate. Finally, if the client wants to decline the 'reserved' dates and times, the reservation status of the diary will be set to be 'free'.

No	Reserve status	Timeout	Permission for a new request
1	Accept	-	No
2	Reserve	No	No
3	Reserve	Yes	Yes
4	Free	-	Yes

Table 6.1: The DASs' time-out

Table 6.1 presents the decision table of the time-out mechanism employed in the DASs' diary. The DASs use it to underpin the decision model about reservation of the DASs' resources in the diary. The table shows the permission status for a new request from the I-Broker to updata the DASs' diary as decribed in the following.

1. There is no time-out if the reservation status of the diary is 'accept'. A new request will not be allowed to update the diary for the dates and times requested. The client who reserved these dates and times is the only one that can decline the booking so that the diary is allowed to be reserved or booked by other requests

2. If the diary is reserved but no time-out has yet occurred, the new request cannot update the diary.

3. However, if the diary is reserved and has 'time-out', the new request can update the diary.

4. If the diary is free, a new request can reserve the diary without checking the time-out status.

6.4 Service Negotiation in CAPTAIN

Figure 6.8 provides a sequence message diagram that illustrates service negotiation in CAPTAIN. It includes all negotiation elements for CAPTAIN supporting the service negotiation between the P-Broker and the DASs to produce the integrated care plan for the client. Pre-negotiation, service negotiation and post-negotiation are the three main stages of negotiation process, explained as in the following.

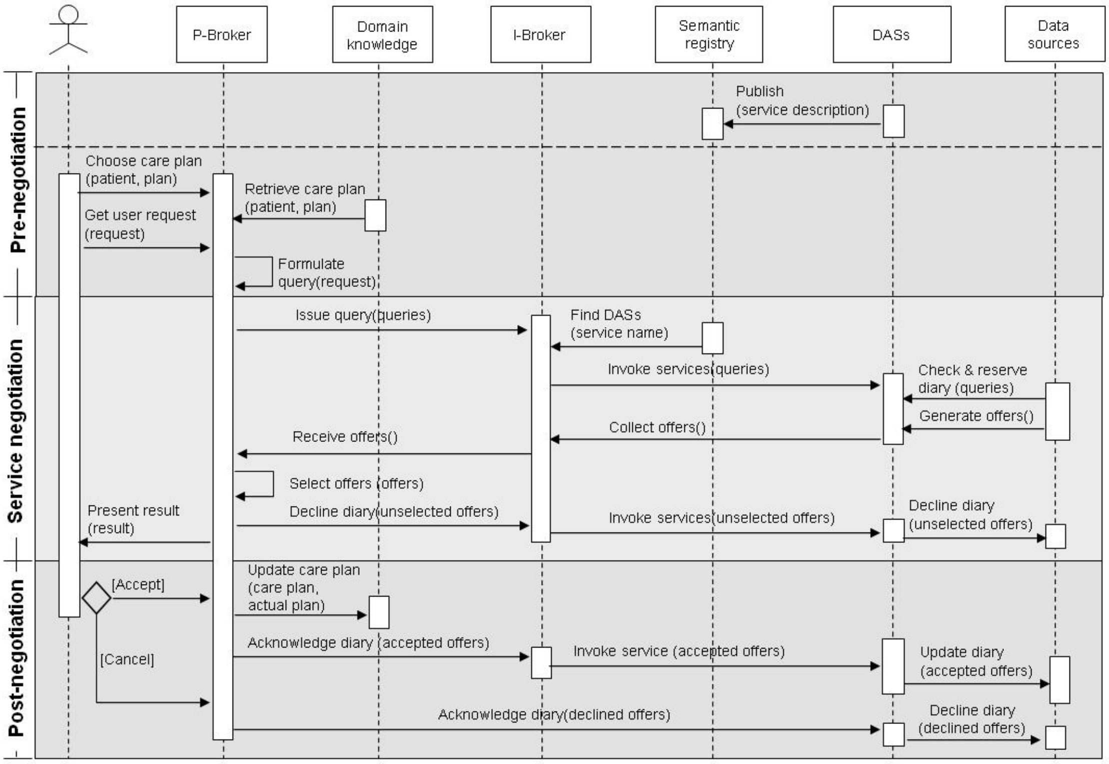

P-Broker
Domain knowledge
I-Broker
Semantic registry
DASs
Data sources
Pre-negotiation
Service negotiation
Post-negotiation
Publish (service description)
Choose care plan (patient, plan)
Retrieve care plan (patient, plan)
Get user request (request)
Formulate query(request)
Issue query(queries)
Find DASs (service name)
Invoke services(queries)
Check & reserve diary (queries)
Generate offers()
Collect offers()
Receive offers()
Select offers (offers)
Present result (result)
Decline diary(unselected offers)
Invoke services(unselected offers)
Decline diary (unselected offers)
[Accept]
Update care plan (care plan, actual plan)
Acknowledge diary (accepted offers)
Invoke service (accepted offers)
Update diary (accepted offers)
[Cancel]
Acknowledge diary(declined offers)
Decline diary (declined offers)

Figure 6.8: The sequence diagram of CAPTAIN

6.4.1 Pre-negotiation

The pre-negotiation stage aims to produce the information needed for the service negotiation stage. It involves three system components of CAPTAIN in this stage: the DASs, the client and the planning broker.

1. The DASs publish their service descriptions to the semantic registry by using a registry service so that the I-Broker can find their service description during service negotiation. The service description of each DAS consists of three basic items of data: service name, service access point and service interface. This data is used by the I-Broker to find the services provided by the DAS, to locate the DAS's location and finally to access its data service respectively.

2. The client generates a request for the planning broker by choosing a care plan that is retrieved from the domain knowledge owned by the P-Broker. Then

the client revises the care plan to tailor it according to the client's needs and preferences together with any constraints. Then the request is summited to the P-Broker to formulate specific queries for the DASs.

3. The care plan formulation of the P-Broker handles the client's request by formulating the client's request as a set of queries for the DASs. Then the care plan implementation receives these queries from the care plan formulation, and submits them to the DASs.

6.4.2 Service Negotiation

Service negotiation occurs between the P-Broker and the DASs. They exchange their messages or negotiation objects through the I-Broker as described below.

1. The care plan implementation service of the planning broker receives the queries from the care plan formulation. It then submits the queries to the information broker.

2. The I-Broker extracts a service name from the queries and uses it to find candidate DASs from the semantic registry. Then it locates and invokes the DASs' data services.

3. Each DASs processes the queries as shown in Figure 6.9.

   (a) Receive and process the queries by formulating the local queries to retrieve data from the data sources

   (b) Check and reserve suitable available resources in its diary to provide the required services for the client

   (c) Set time-out period for the reserved resources

   (d) Generate the corresponding offers

(e) Send the offers to the I-Broker

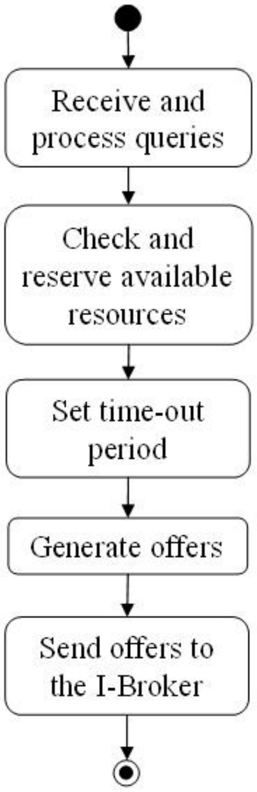

Receive and process queries
Check and reserve available resources
Set time-out period
Generate offers
Send offers to the I-Broker

Figure 6.9: Processing queries in the DASs

4. The I-Broker collects the offers from the DASs and sends them to the P-Broker.

5. The P-Broker receives the offers from all DASs and processes them as illustrated in the Activity diagram shown in Figure 6.10.

   (a) Select and accept the offers that meet the client's needs and preferences as well as constraints

   (b) For the unselected offers, send 'decline' messages back to the DASs

   (c) Generate or modify an integrated care plan and present the results to the client

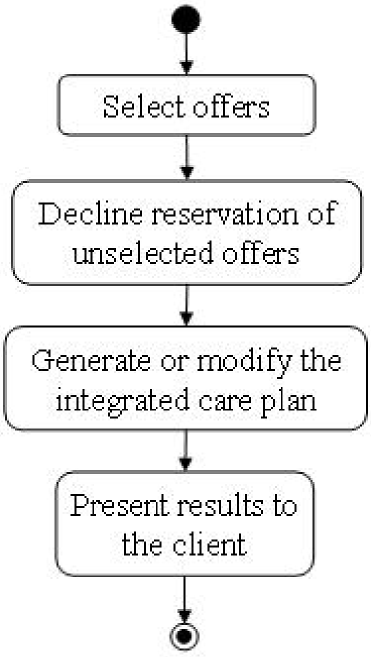

Select offers
Decline reservation of unselected offers
Generate or modify the integrated care plan
Present results to the client

Figure 6.10: Processing offers in the P-Broker

6. The DASs update their records according to whether services are accepted or declined. (Should no response be received when a given time-out period, any offers are assumed to have been declined.)

6.4.3 Post-negotiation

The post-negotiation phase involves the decision-making by the client and the subsequent updating of the DASs' data sources. The client needs to decide whether to accept or decline the result from the P-Broker. Then the P-Broker will send an acknowledgement of the decision-making to the DASs. Based on this, the DASs update the status of the client's reserved offers in the diaries, as being accepted or declined respectively.

6.5 Implementation

This section describes the implementation of the CAPTAIN application. It starts from the original prototype that was developed using IBM WebSphere software development tools. Then the final version of the CAPTAIN application will be described by showing the user interfaces that represent the three main operations related to the creation and modification of the integrated care plan.

6.5.1 The Original Prototype

The original prototype was built using IBM WebSphere Studio Application Developer Version: 5.1.1, working on Windows platform. IBM WebSphere provides Web services development tools that are based on open, cross-platform standards: SOAP, UDDI and WSDL. It supports Java 2 Platform, Enterprise Edition (J2EE) that supplies a component-based approach to the design, development, assembly, and deployment of enterprise applications.

The prototype had provided adequate support for IBHIS. The experience from this led to the initial development of this new service-oriented application called CAPTAIN. However, since WebSphere had not evolved adequately, the CAPTAIN application was then re-engineered based on Eclipse IDE for Java EE Web Developers and the Apache Tomcat web application server, that both work on the Windows platform. There are several advantages of developing a service-oriented application with the Eclipse IDE. For example, it is open source and supports the latest of Java and Web services technologies. The application within the original prototype was therefore migrated to the platform of Eclipse IDE. Then new functions were created so that the new application could perform as needed to meet the purpose of this research.

6.5.2 The CAPTAIN Application

The CAPTAIN system can be accessed remotely through any Web browser. There are three main types of client operations involved in creating and managing the integrated care plan: choosing an operation, generating a request and choosing offers. These operations are described as follows.

1. **Choosing an operation**

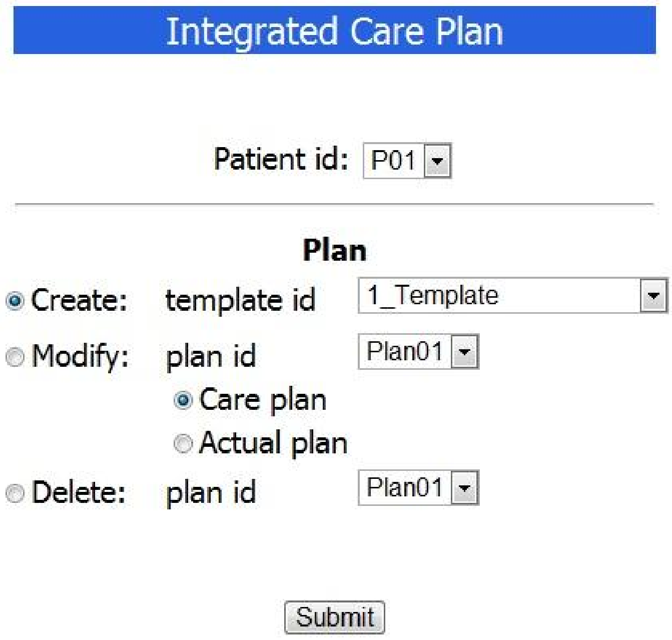
   Figure 6.11 shows the user interface. The client chooses a patient id and the following operations related to the integrated care plan.

   - *"Create"*

     The client selects the care plan template that is suited for the patient in order to create the integrated care plan for the patient.

   - *"Modify"* and *"Delete"*

     The client selectes the care plan and then the types of the care plans, a care plan or an actual plan, to be modified and deleted respectively.



Integrated Care Plan

Patient id: P01

Plan

● Create: template id 1_Template

○ Modify: plan id Plan01

● Care plan

○ Actual plan

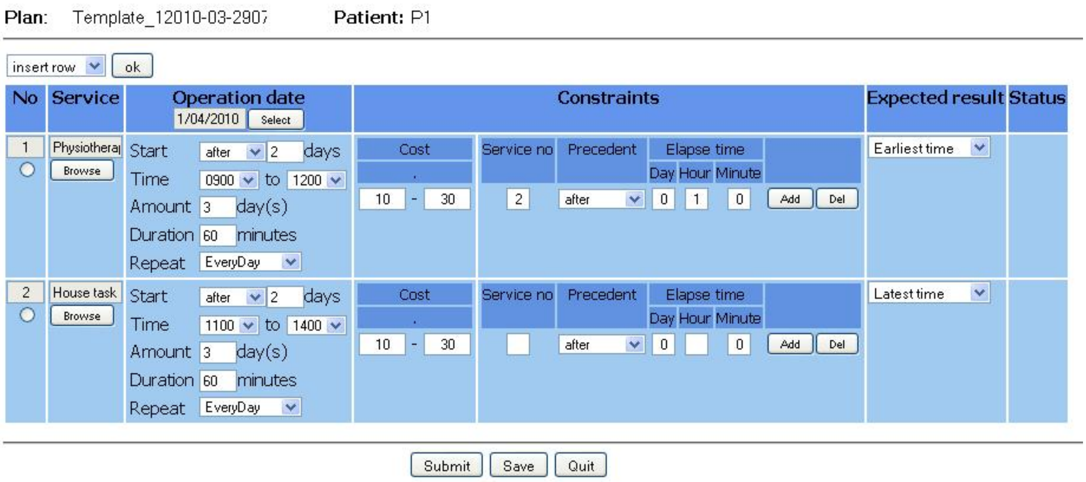
○ Delete: plan id Plan01

Figure 6.11: Choosing an operation of the CAPTAIN application

2. **Generating a request**

Figure 6.12 shows the user interface that the client uses to generate requests from the care plan template, the care plan or the actual plan for the P-Broker. The client performs two main operations as follows.

- Generate the client's requests by revising the details of the care plan that consists of four main parts: service name, details of dates and times, constraints and expected result
- Submit the requests to the P-Broker so that it uses the requests to negotiate with the DASs in order to to create or modify the integrated care plan for the client



Plan: Template_12010-03-2907 **Patient:** P1

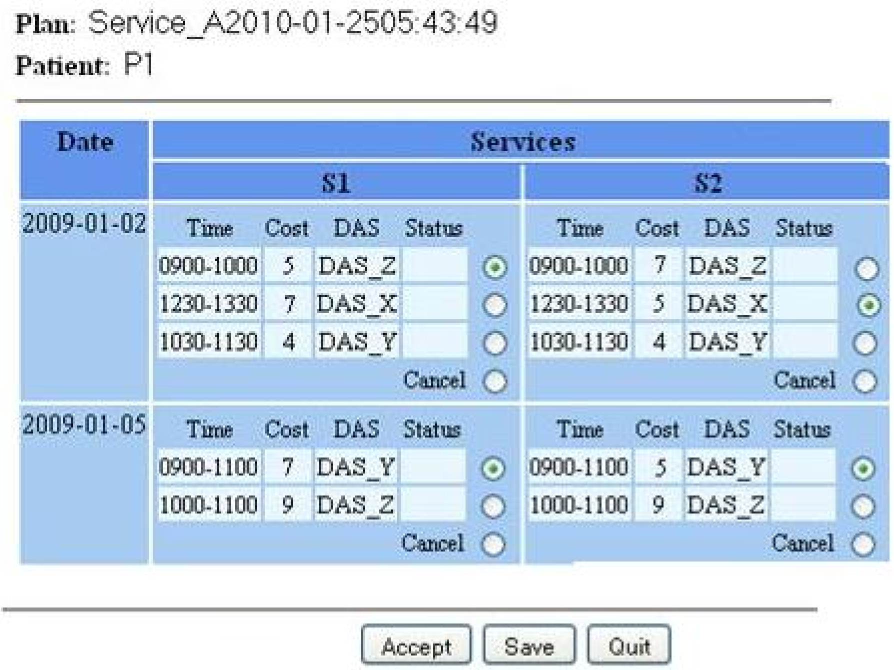
insert row ok

No	Service	Operation date 1/04/2010 Select	Constraints	Expected result	Status
1 ○	Physiotheraı Browse	Start after 2 days Time 0900 to 1200 Amount 3 day(s) Duration 60 minutes Repeat EveryDay	Cost: 10 - 30 Service no: 2 Precedent: after Elapse time: Day 0, Hour 1, Minute 0 Add Del	Earliest time	
2 ○	House task Browse	Start after 2 days Time 1100 to 1400 Amount 3 day(s) Duration 60 minutes Repeat EveryDay	Cost: 10 - 30 Service no: Precedent: after Elapse time: Day 0, Hour , Minute 0 Add Del	Latest time	

Figure 6.12: Generating a request of the CAPTAIN application

3. **Choosing offers**

Figure 6.13 shows the results from the P-Broker. They are offers provided by the DASs. The P-Broker chooses a group of offers that are preferred before presenting the results to the client. The client will make his/her choice before accepting the results. Then the P-Broker will send an acknowledgement about both the selected and unselected offers to the DASs, so that they can update their diaries. The accepted care plan will be kept or updated as part of the domain knowledge of the P-Broker.



Plan: Service_A2010-01-2505:43:49
Patient: P1

Date	Services									
	S1					S2				
2009-01-02	Time	Cost	DAS	Status		Time	Cost	DAS	Status	
	0900-1000	5	DAS_Z		●	0900-1000	7	DAS_Z		○
	1230-1330	7	DAS_X		○	1230-1330	5	DAS_X		●
	1030-1130	4	DAS_Y		○	1030-1130	4	DAS_Y		○
				Cancel	○				Cancel	○
2009-01-05	Time	Cost	DAS	Status		Time	Cost	DAS	Status	
	0900-1100	7	DAS_Y		●	0900-1100	5	DAS_Y		●
	1000-1100	9	DAS_Z		○	1000-1100	9	DAS_Z		○
				Cancel	○				Cancel	○

Figure 6.13: Choosing offers of the CAPTAIN application

6.5.3 Development and Deployment Tools

The CAPTAIN application has been designed and built based on experiences from IBHIS in order to meet the objective of this research. IBHIS has demonstrated that it is possible to employ Web Services technologies from several aspects to develop a service-oriented application which works.

System development and deployment tools

The service-oriented application of CAPTAIN has been based on Java 2 Enterprise Edition (J2EE) Web Services technologies. It is developed and run within Eclipse Java EE IDE for Web Developers environment working on Windows platform. It supports current Web services standards such as SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration) and XML (Extensible Markup Language).

The CAPTAIN application is deployed on the Apache Tomcat web application server. Besides of the web user interface and the semantic registry, all of the system components of CAPTAIN (the P-Broker, the I-Broker, the DASs and registry service) are deployed as Web Services. Therefore, they can work together through the three basic platform elements of Web Services: SOAP, WSDL and UDDI.

Web user interface

The CAPTAIN application is a dynamic web application. The web user interface for the client and the P-Broker were developed and implemented by using HTML pages, JavaServer Pages (JSP) and Java Servlets. These can be used to build a complete Java J2EE Web Application, including a front-end and a back-end to deal with the dynamic and complicated tasks of the client and server sides respectively.

XLS (Extensible Stylesheet Language) has also been used to present the results from the P-Broker to the client via the web user interface. It is a transformation language that is used to describe or specify rules about how an XML document is transformed, formatted or displayed. It converts the XML documents received from the P-Broker by using XPath, XSL-FO (XSL Formatting Objects) and XSLT (XSL Transformations) to provide an HTML page. XSL-FO is used to format the XML documents. XSLT transforms the XML document into other forms such as HTML that appear in a web browser to a client.

DASs

The DASs' services are employed and described by using WSDL. Their services are invoked by the client through WSDL. The DASs' data sources store data and information based on a MySQL DBMS platform that is a multithreaded and multiuser SQL database management system (DBMS). The data and information of the domain knowledge and the DASs are mainly stored in the format of XML documents so that they can provide a sophisticated care plan and a set of heterogeneous and distributed data sources.

Semantic registry

The semantic registry runs on the Apache jUDDI (an open source Java implementation of the Universal Description, Discovery, and Integration), allowing clients to publish and to discover services dynamically. The DASs use the registry service to publish their service descriptions to the semantic registry. Then the I-Broker finds the DASs by their service descriptions from the semantic registry. So the services provided by the DASs can be located and invoked by the information broker.

System communication

The communication among CAPTAIN's system components uses messaging in the form of XML-based documents. Their messages are exchanged as SOAP document style messages. The messaging style used within the prototype is based on the Remote Procedure Call (RPC) which is a request/response-based synchronous communication. With this method, the client, especially the I-Broker, can interact with a server or invoke services through its proxies until the server side replies to the request.

6.6 Summary

This section introduces a service-oriented broker application, called the CAPTAIN broker. The aim of the CAPTAIN broker is to provide care planning services to a client. It consists of two main service brokers: the care planning broker or P-Broker;

and the information broker or I-Broker. The P-Broker provides care planning services for the client. It acts on behalf of the client in negotiating with various service providers to produce the integrated care plan that meets the needs of the client. The I-Broker performs as an intermediary between the P-Broker and the service providers during service negotiation. The I-Broker finds and locates relevant providers of services, and organising exchange of the messages between the negotiation parties.

The CAPTAIN broker includes a negotiation model consisting of the negotiation process, negotiation object, negotiation protocol and decision model. The negotiation model is used by the negotiation participants to deal with the negotiation situation that arise when creating a care plan. The result of negotiation is the integrated care plan that should meet the needs of the client.

The CAPTAIN application was developed and run based on Web Services technologies. It supports current Web Services standard, for instance SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery, and Integration) and XML (Extensible Markup Language). The CAPTAIN application is a dynamic web application. All main system components of CAPTAIN, except the semantic registry, are deployed as Web Services.

Chapter 7

Evaluation

This chapter presents an evaluation of CAPTAIN and reviews the ‘proof of concept’ results from using CAPTAIN. A Use Case and a number of related scenarios are used for the evaluation of CAPTAIN in order to assess the different aspects of its main features.

7.1 Introduction

Table 7.1 shows the main features of CAPTAIN that have been exercised in order to evaluate the system. The evaluation of each feature will be described in more detail in Section 7.2.

These scenarios used for evaluation in Section 7.2 are all derived from Edith’s Use Case, as specified in Section 5.4.1 of Chapter 5. To perform the evaluation of CAPTAIN, there are three service providers that are represented as DASs. Each DAS contains data and information related to health care services provision for a patient. They publish their service descriptions to the semantic registry so that the I-Broker can find their service description and invoke the DASs’ data access services. Each DAS has a diary that is used for reserving and booking the services that it provides for the patient. The diary contains data such as date, time and resources, together with values for any timeouts used when reserving and booking slots in the diary.

Feature	Scenario		Request	Initial state of DASs	After negotiation: expected results
Function	1	Creation of a care plan	A care plan template	Diary with resources being available to provide services	A new care plan
	2	Modification of the care plan	A revised care plan	Diary with some resources reserved and booked	The modified care plan
	3	Management of Appointment Conflict	Requests for a new service	Diary with the service appointments conflicting with the request	Diary with appointments rearranged after resolving the conflict
	4	Management of the Parallel Access and Timeout	Requests for a new service	Diary with the service appointments that has timeout and conflicts with the request	Diary with appointments rearranged after resolving the conflict and timeout
Data access	5	Role of read and write data sources	Service request and client's response to providers' offers	Diary with resources being available to provide services	Diary updated to a new service request
Negotiation	6	Time-out of reservation and booking services	Requests for reservation and booking services	Diary with resources being available to provide services	Diary updated to a new service request
	7	Proposition and Selection Offers of the DASs and the P-Broker	Request for reserving and booking service	Diary with some resources being available for reserving and booking service	Multiple offers selected by the P-Broker for the client
	8	System observation	The care plan template	Diary with resources being available to provide services	The ordering of events occuring in CAPTAIN

Table 7.1: Scenarios for evaluation of CAPTAIN's main features

The set of scenarios shown in Table 7.1 are intended to demonstrate the three main features of the CAPTAIN application from various aspects: function, data access and negotiation. These features are described as follows.

1. The *functions* of the CAPTAIN broker

   The CAPTAIN broker or the P-Broker provides a care planning service that performs as the central element of the CAPTAIN system. The client uses and operates the system, through the CAPTAIN broker, to create, modify and manage the integrated care plans. These operations are evaluated by the set of scenarios 1 to 4.

2. The role of *data access* (read-write)

   When the client creates or modifies the integrated care plan and submits a request to the service providers or the DASs, the DASs deal with the client's request by reading and changing data in their diaries in order to negotiate with the client. The DASs access the data sources to read and write data records during and after the service negotiation. Therefore, scenario 5 is used to evaluate the role of data access (read-write).

3. The *negotiation* model for CAPTAIN

   The negotiation model is the key part of the service negotiation involving all of the main system components of CAPTAIN. During service negotiation, the negotiation elements have to perform properly to meet the purposes of the negotiation participants. The negotiation elements of the negotiation model are evaluated as follows.

   - Scenarios 6 and 7 evaluate the decision models used by the P-Broker and the DASs. They are also demonstrate the read and write features which are related to the decision-making of the DASs.
   - Section 8 examines the negotiation model for CAPTAIN, and so focuses on the negotiation object, negotiation protocol and negotiation process.

7.2 Scenarios

The evaluation is based on several scenarios that investigate its operations when viewed from different aspects. Edith's Use Case as specified in section 5.4.1 has been used in order to investigate CAPTAIN. Thus, each scenario describes and discusses the use and result of a specific application of CAPTAIN.

7.2.1 Creation of a Care Plan

Case

The consultant assesses Edith for her needs and constraints. Edith will need physiotherapy for her injury that can be performed at a community hospital. Nevertheless, she will need ambulance transport from her home to the community hospital. Ambulance transportation is provided by several transport providers. Then the consultant passes the result of Edith's assessment to the nurse to create an integrated care plan for her treatment and support after being discharged from the rehabilitation ward.

Operation

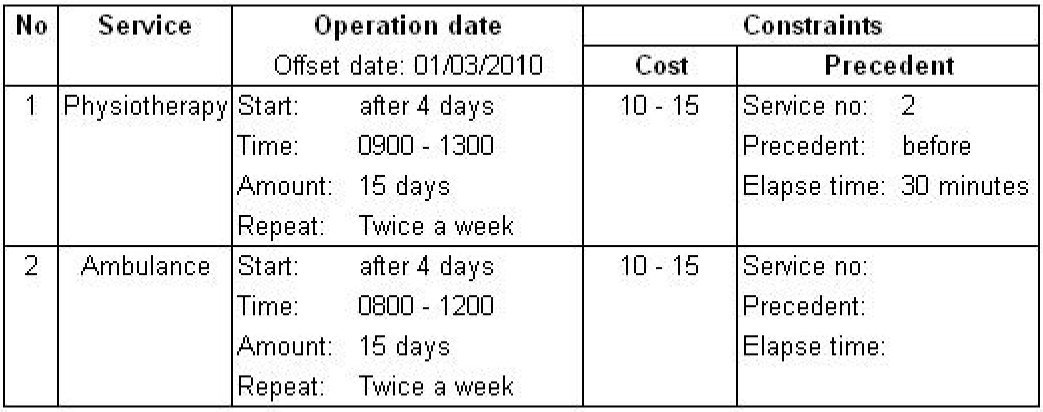
The nurse uses the CAPTAIN system to create the integrated care plan for Edith as in the following sequence.

1. Retrieve and choose a care plan template that is suited for Edith's case
2. Change the details of the template by identifying the needs, preferences and constraints of Edith
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that met Edith's needs

5. Accept the selected offers and save the new care plan together with its actual plan to the system's database

Result

The care plan template is a pre-defined care package that is used to create a request for creating a new care plan. The template can be changed by a client. Its structure is based on the client's needs and any constraints that are applied when negotiating with the providers. This implies that the template consists of several and different independent variables whose value can be changed by the client.



No	Service	Operation date	Constraints	
		Offset date: 01/03/2010	Cost	Precedent
1	Physiotherapy	Start: after 4 days Time: 0900 - 1300 Amount: 15 days Repeat: Twice a week	10 - 15	Service no: 2 Precedent: before Elapse time: 30 minutes
2	Ambulance	Start: after 4 days Time: 0800 - 1200 Amount: 15 days Repeat: Twice a week	10 - 15	Service no: Precedent: Elapse time:

Table 7.2: The example of a request for creating a new care plan

Table 7.2 provides an example of the care plan template that is used to create a request for generating the new careplan. The structure of Table 7.2 consists of four main parts.

1. Service number

   A service number represents a number of the requested service that may be referenced by the other relevant services.

2. Service name

   A service name is identified the client to request for the service offers from the providers.

3. Operation date

   This part is used to specify the dates and times of the services requested by the patient as follows.

   (a) Start and offset dates

       Each service has to be defined in terms of the number of days that a ‘*Start*’ date begins ‘before’ or ‘after’ the ‘*Offset date*’. The ‘Offset date’ is the time at which the operation date of the care plan is supposed to begin. All services identified in the integrated care plan have the same ‘Offset date’ showing how many days the service has taken from the begining of the care plan.

   (b) Time and frequency

       The client defines the following details of service provision to be supplied by the providers.

       - ’*Time*’ is the range of possible times.
       - ’*Amount*’ is the number of days.
       - ’*Repeat*’ is the frequency of service provision.

4. Constraints

   This part identifies any constraints preferred by the patient for the service offered by the providers. For the purpose of this study, there are two types of constraints addressed in the template.

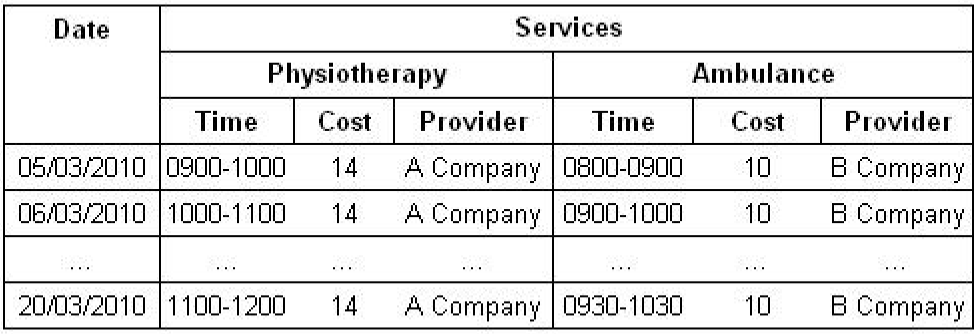
   (a) Cost

       The range of the acceptable cost is identified by the patient and issued to the providers so that they can propose service offers with acceptable cost to the patient (or decline to make an offer).

   (b) Precedent

The precedent is specified when there is a ‘happened before’ relationship between two services. For the example of request in Table 7.2, Edith needs ambulance transportation from her home to the community hospital about 30 minutes before her physiotherapy treatment is taken.

After submitting the request of the client to the P-Broker for the care planning services, the service negotiation process seeks to find agreement between the client’s needs and the providers’ constraints. The results from the providers are then the offers that are selected by the client.



Date	Services					
	Physiotherapy			Ambulance		
	Time	Cost	Provider	Time	Cost	Provider
05/03/2010	0900-1000	14	A Company	0800-0900	10	B Company
06/03/2010	1000-1100	14	A Company	0900-1000	10	B Company
...	...	...	...	...	...	...
20/03/2010	1100-1200	14	A Company	0930-1030	10	B Company

Table 7.3: An example result of creating the new care plan

Table 7.3 presents an example of the offers selected by the client. It is composed of the columns of dates and services offered by the providers. Each date identifies the services provided by the providers on that day. Each service presents detail about time, cost and the name of the provider.

Finally, the client acknowledges the providers, via the P-Broker, with the client’s decision about both selected and unselected offers . Consequently, the request and result tables, as shown in Table 7.2 and 7.3, are saved to the system’s domain knowledge as the new care plan and the actual care plan respectively.

7.2.2 Modification of the Care Plan

Case

After three weeks of the physiotherapy treatment, the senior physiotherapist assesses Edith and finds that she needs assistance to cope with daily living at home, due to her reduced mobility. Her daughter cannot come to assist her with bathing for a few weeks, which she normally does once a week. She also needs someone to clean her house too. Hence, the nurse has to plan for a nursing aide and community services to undertake the house tasks until her daughter comes back.

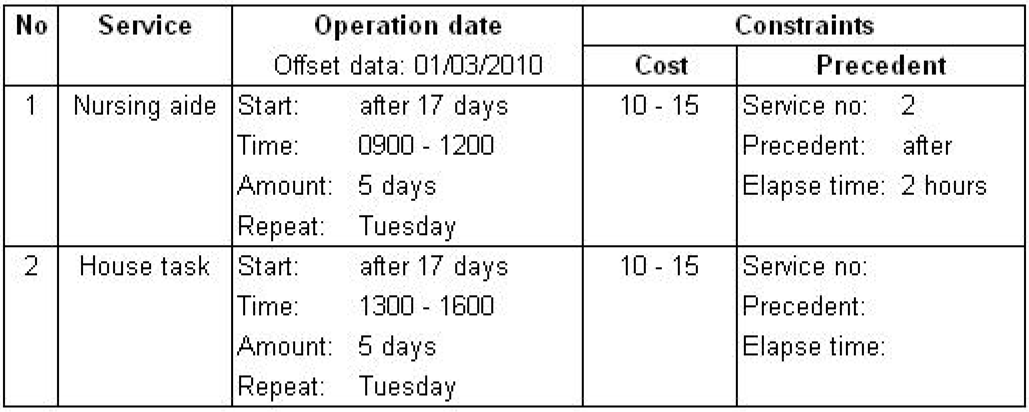
Operation

The nurse uses the CAPTAIN system to modify the integrated care plan for Edith as follows.

1. Retrieve Edith's care plan from the system's database
2. Change the details of the care plan
   (a) Cancel the physiotherapy and ambulance transport
   (b) Add the new requirements for a nursing aide and house task to the care plan.
3. Issue the requests for the cancelled and new services to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and update the care plan to the system's database

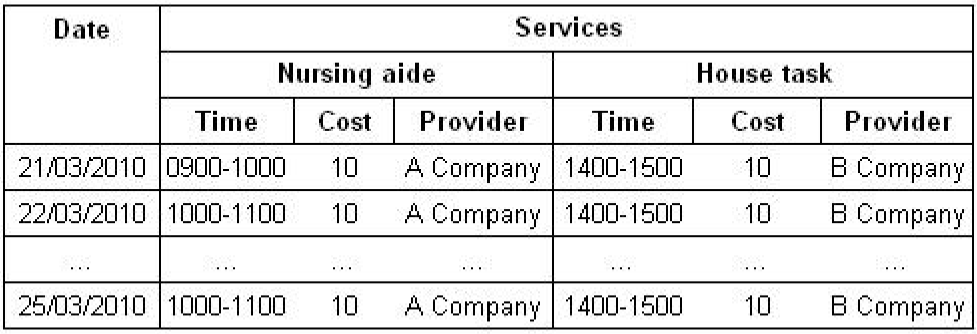
Result

Table 7.4 is the example of the request for modifying the care plan for Edith that is retrieved from the system's domain knowledge and modified by the client. The request is issued to the providers. The providers then return their offers, that are used by the client to manage their selection, as presented in Table 7.5.



No	Service	Operation date	Constraints	
		Offset data: 01/03/2010	Cost	Precedent
1	Nursing aide	Start: after 17 days Time: 0900 - 1200 Amount: 5 days Repeat: Tuesday	10 - 15	Service no: 2 Precedent: after Elapse time: 2 hours
2	House task	Start: after 17 days Time: 1300 - 1600 Amount: 5 days Repeat: Tuesday	10 - 15	Service no: Precedent: Elapse time:

Table 7.4: The example of request for modifying the care plan



Date	Services					
	Nursing aide			House task		
	Time	Cost	Provider	Time	Cost	Provider
21/03/2010	0900-1000	10	A Company	1400-1500	10	B Company
22/03/2010	1000-1100	10	A Company	1400-1500	10	B Company
...	...	...	...	...	...	...
25/03/2010	1000-1100	10	A Company	1400-1500	10	B Company

Table 7.5: The example of result of modifying the care plan

The modification of the care plan begins from the care plan that was created in the creation stage. The modification process follows a similar process to that used in the creation of the care plan. The client modifies the care plan, and submits the request to the P-Broker. The P-Broker formulates a new request and issues it to the DASs. After that, it receives the offers from the DASs and selects from them according to the client's needs. Finally, the P-Broker sends the results to the client to make the final decision.

From the service negotiation point of view, the negotiation participants may need to negotiate if both sides have a conflict of interest or if they do not achieve mutual agreement on the terms and conditions. The modification of the care plan occurs when the needs and conditions of the patient deviate from the existing care plan. Hence, the client has to renegotiate with the service providers to agree the new terms of agreements and conditions.

7.2.3 Management of Appointment Conflict

Case

The care planner has made appointments for the house task and physiotherapy treatment on Tuesday and Thursday afternoons for Edith respectively. Before the appointment dates, the consultant contacts and requests the district nurse to visit and assess Edith's progress. However, the district nurse can be available only on Tuesday and Thursday afternoons because she already has the appointments on the other days. The district nurse normally visits a patient in the afternoon because she has the other tasks to do at a healthcare centre and a community hospital. Moreover, only a limited number of district nurses are available to provide healthcare services. As a result, this causes a conflict of appointments needing a rearrangment of the appointments.

Therefore, the care planner has to modify Edith's care plan based on two criteria: case priority and care services sequence. For Edith's case, the order of case priority from high to low is the physiotherapy, the district nurse and the house task. Then the planner considers the conflict of appointments regarding the district nurse, care services and house tasks. The planner has to make an appointment for the district nurse on Tuesday afternoon, then has to find an other available time for the house task.

Operation

The care planner uses the CAPTAIN system to rearrange the appointments of care services for Edith in the integrated care plan. These are the operations followed by the planner.

Add the house task and for physiotherapy treatment service

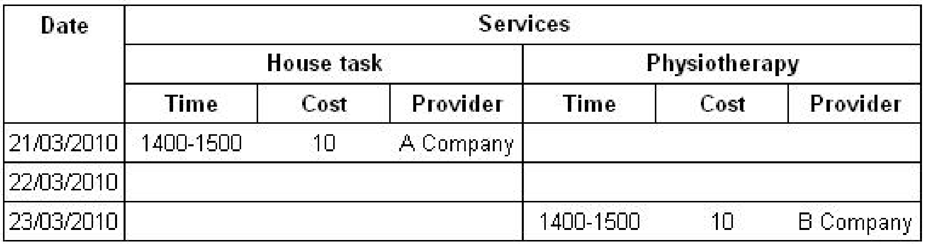
1. Retrieve Edith's care plan from the domain knowledge
2. Change the details of the care plan by adding the house task and physiotherapy treatment services for Edith on Tuesday and Thursday afternoon
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and save the care plan to the domain knowledge

Add the district nurse service

1. Retrieve Edith's care plan from the domain knowledge
2. Change the details of the care plan by adding the district nurse service for Edith on the same week with the house task and physiotherapy treatment services
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and save the care plan to the domain knowledge

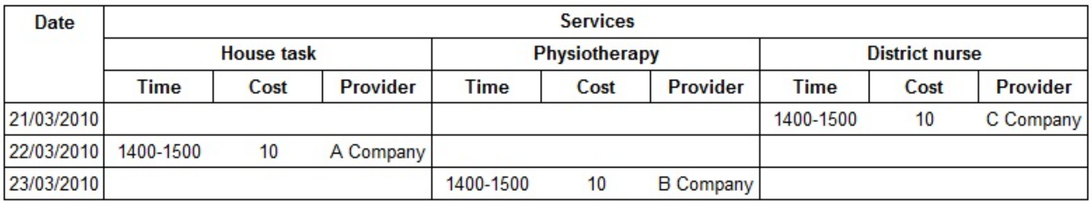
Result

Tables 7.6 and 7.7 illustrate the example of the appointments before and after managing the appointment conflict for Edith.



Date	Services					
	House task			Physiotherapy		
	Time	Cost	Provider	Time	Cost	Provider
21/03/2010	1400-1500	10	A Company			
22/03/2010						
23/03/2010				1400-1500	10	B Company

Table 7.6: The example of appointments before managing appointment conflict



Date	Services								
	House task			Physiotherapy			District nurse		
	Time	Cost	Provider	Time	Cost	Provider	Time	Cost	Provider
21/03/2010							1400-1500	10	C Company
22/03/2010	1400-1500	10	A Company						
23/03/2010				1400-1500	10	B Company			

Table 7.7: The example of appointments after managing appointment conflict

The management of an appointment conflict involves the addition and the cancellation of services in the integrated care plan. The conflict of appointments occurs when there are more than one service wanting to make an appointment at the same time. However, there must be one service that is most important to the patient that can be made the appointment, while the rest of the services have to give up their appointments and to find the other appropriate dates and times for making the appointments.

The management of the appointment conflict is performed as in the following main operations.

1. The P-Broker receives the request from the client and formulates the request into appropriate forms of queries. The P-Broker, then, issues the queries to the DASs for offers. When the P-Broker receives the offers from the DASs, it

will select the offers that best meet the client's needs, and the dates and times must be available for those selected offers.

2. The P-Broker retrieves the information of the current appointments specified in the actual care plan and of services' priority from the domain knowledge. Then the P-Broker employs the appointment conflict function of the decision model to analyse the offers and the current appointments, as well as services' priority, to select the appropriate appointments for the client.

3. The P-Broker uses the following strategy to select the offers to arrange the conflicted appointments,

    - Cancel the appointment of the house task, lower priority service
    - Find another available time and making a new appointment for the house task
    - Make the appointment for the district nurse at the identified time requested by the client
    - Present the result to the client to make final decision

7.2.4 Management of Parallel Access and Timeout

Case

The nurse makes an appointment for the district nurse on Tuesday for Edith. However, the consultant informs her later that she will need a physiotherapy treatment. She also needs someone to clean her house. Then, she asks the care planner to find service providers and to make appointments with them for her. The care planner modifies the care plan for her by issuing the requests to the service providers for the physiotherapy treatment and house task services on the same week with the appointment for the district nurse. However, the service providers make the offers but the dates and times are the same as the district nurse appointment. The appointments must be rearranged by considering the case priority needed by the patient. For this case, the physiotherapy appointment is the highest priority. Then the district nurse and the house task are ordered from high to low priority respectively. However, only a limited number of physiotherapists are available to provide healthcare service. There is a timeout for reserving the physiotherapist. So, the care planner has to confirm the provider within the timeout, otherwise the reservation for the physiotherapist will be cancelled. Then, the care planner has to modify Edith's care plan to resolve the conflicts that are based on two criteria: case priority and care services sequence.

Operation

The care planner uses the CAPTAIN system to rearrange the appointments of care services for Edith in the integrated care plan. These are the operations followed by the planner.

Add the district nurse service

1. Retrieve Edith's care plan from the domain knowledge
2. Change the details of the care plan by adding the district nurse service for Edith on Tuesday afternoon
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and save the care plan to the domain knowledge

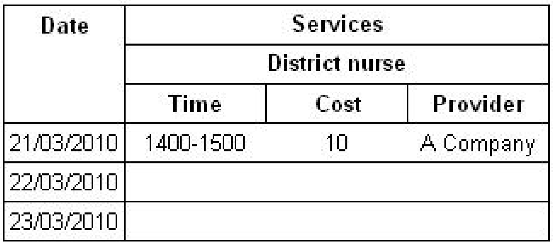
Add the physiotherapy treatment and house task services

1. Retrieve Edith's care plan from the domain knowledge

2. Change the details of the care plan by adding the physiotherapy treatment and house task services for Edith
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and save the care plan to the domain knowledge after the physiotherapy treatment's timeout

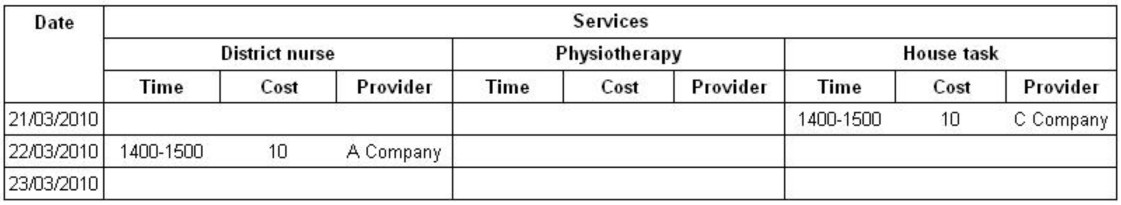
Result

Tables 7.8 and 7.9 illustrate the example of the appointments before and after managing the appointment conflict for Edith.



Date	Services		
	District nurse		
	Time	Cost	Provider
21/03/2010	1400-1500	10	A Company
22/03/2010			
23/03/2010			

Table 7.8: The example of appointments before managing appointment conflict



Date	Services								
	District nurse			Physiotherapy			House task		
	Time	Cost	Provider	Time	Cost	Provider	Time	Cost	Provider
21/03/2010							1400-1500	10	C Company
22/03/2010	1400-1500	10	A Company						
23/03/2010									

Table 7.9: The example of appointments after managing appointment conflict

This scenario illustrates the conflicts of appointments when multiple services need to make the appointments at the same time. The client submits the request to the P-Broker to formulate queries for the DASs. The P-Broker, then, issues the queries to the DASs. When the P-Broker receives offers from the DASs, it will select the offers that best meet the client's needs.

The management of parallel access and timeout involves the following main operations.

1. When there is a conflict between the appointments, the P-Broker retrieves the information of the current appointments specified in the actual care plan and of services' priority from the domain knowledge.
2. The P-Broker employs the appointment conflict function of the decision model to analyse the offers and the current appointments, as well as services' priority, to select the appropriate appointments for the client.
3. The P-Broker also uses the strategy to make the appointment by re-organising the services based on priority, as follows.
    - Cancel the appointment for the district nurse, priority lower than the physiotherapy treatment.
    - Make the appointment for the physiotherapy treatment at the identified time requested by the client.
    - Find available time and makes a new appointment for the district nurse and the house task respectively.
    - Present the result to the client to make final decision. However, the care planner accepts the result later than the physiotherapy treatment's timeout. So the provider has to cancel its reservation, and the care planner has to make a appointment for the physiotherapy session.

7.2.5 The Role of Read and Write Data Sources

Case

The consultant wants to see Edith in 6 weeks in the outpatient clinic after the physiotherapy treatment in order to assess her needs with the senior physiotherapist.

So, the nurse modifies the care plan to include a meeting with the consultant at the clinic that is near her home.

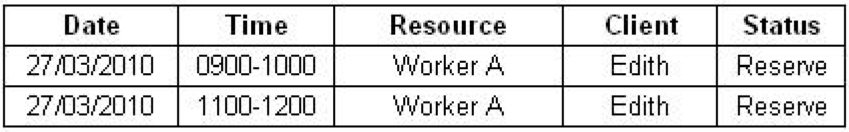
Operation

The nurse uses the CAPTAIN system to modify the integrated care plan for Edith as in the following.

1. Retrieve Edith's care plan from the system database
2. Add a new service into the care plan and change its service details according to her needs
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and save the new care plan to the system's database

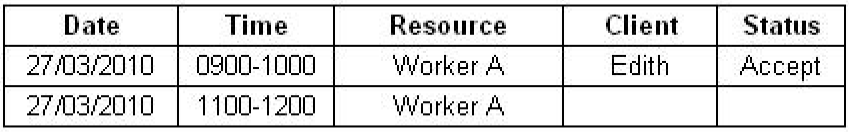
Result

Table 7.10 illustrates the examples of the reservation DAS's diary, in response to the client's request. The DAS updates its diary by reserving the date, time and resource of service that is avialable for the client's request. Table 7.11 presents the examples of bookings in the diary after the DAS receives as 'accept' response from the client.



Date	Time	Resource	Client	Status
27/03/2010	0900-1000	Worker A	Edith	Reserve
27/03/2010	1100-1200	Worker A	Edith	Reserve

Table 7.10: Reserve DAS's diary



Date	Time	Resource	Client	Status
27/03/2010	0900-1000	Worker A	Edith	Accept
27/03/2010	1100-1200	Worker A		

Table 7.11: Booking DAS's diary

The reservation and booking of the diary slots is based on the reservation states of the given DASs' diary. As specified in Section 6.7, there are three types of states: 'free', 'reserve' and 'accept'. The reservation and booking states of the diary for the specific date and time are toggled in different states for different situations as in the following.

- The initial state of reserve status is 'free'.
- The diary slot is reserved ('reserve'), when there is a request from the client demanding a service at the specific date and time, and the diary slot for that date and time is in a 'free' state. Then, the offer will be generated and returned to the client to make the decision as to whether to accept the offer or not.
- The diary will be updated, changing the status to 'accept' or 'free' if the client accepts or declines the offer respectively.

The role of data access for CAPTAIN is therefore in an active role (read-write). The data sources in the DASs are accessed according to the requests from the P-Broker. They are read and written to during and after service negotiation between the P-Broker and the DASs. Their negotiation is based on content.

From the perspective of the DASs as service providers, they can negotiate with the client or service requestor by changing or updating the data in their data sources so that they can generate different kinds of offers to the client. On the other hand, the client can also negotiate with the DASs by responding the client's decision making based on the DASs' offers.

7.2.6 Timeout of Reservation and Booking Services

Case

The senior physiotherapist assesses Edith and finds that she does not need the physiotherapy treatment any more. However, she still needs assistance to cope with daily living at home once a week for two weeks, due to her reduced mobility. Her daughter, Penny, wants to come to assist her with bathing and house cleaning on Tuesday during the two weeks, but she is not sure whether she can come because of personal needs. So she asks the nurse to find and reserve for a nursing aide and house tasks from community nursing services to undertake this. Then she will tell the nurse later about her decision on this.

Her daughter tells the nurse a few days later that she can not come to assist Edith by herself on the days that both services have been reserved. Then she asks the nurse to book the offers reserved for both services so that there will be someone to take care of her mother. However, the reserved nursing aide service has already been booked by the other client because of its time-out. The providers do not have many resources for providing the nursing aide service. Therefore, the reservation for this service has timed-out. If the client does not confirm the reserved service within the time-out period, the service will be set to be avialable for another client to reserve or book the service.

Operation

The nurse uses the CAPTAIN system to modify the integrated care plan for Edith as in the following.

Reserve the services

1. Retrieve Edith's care plan from the system database
2. Modify the care plan
   (a) Add the nursing aide and house cleaning services into the care plan
   (b) Change service detail of both services according to Edith's needs
3. Issue the requests to the P-Broker

4. Receive and select the providers' offers that meet Edith's needs

5. Save the care plan and actual plan to the system's database

Book the services

1. Retrieve Edith's actual plan from the system database

2. Choose to accept the reserved offers for the nursing aide and house cleaning services

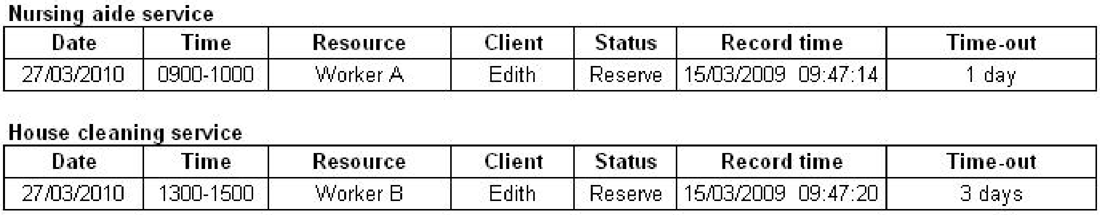
3. Issue the requests to the P-Broker

4. Receive the results from the providers

5. Accept and save the actual plan to the system's database

Result

Table 7.12 illustrates the examples of the DAS's diary slots that are reserved by the client. Table 7.13 presents the examples of bookings in the diaries after the DAS receives confirmation of the booking from the providers reserved offers.



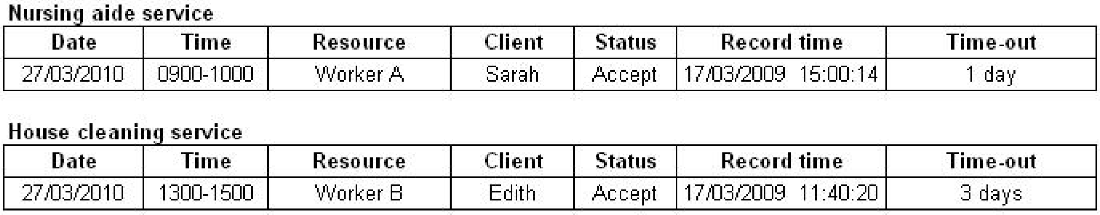
Nursing aide service

Date	Time	Resource	Client	Status	Record time	Time-out
27/03/2010	0900-1000	Worker A	Edith	Reserve	15/03/2009 09:47:14	1 day

House cleaning service

Date	Time	Resource	Client	Status	Record time	Time-out
27/03/2010	1300-1500	Worker B	Edith	Reserve	15/03/2009 09:47:20	3 days

Table 7.12: The example of diary slots before time-out



Nursing aide service

Date	Time	Resource	Client	Status	Record time	Time-out
27/03/2010	0900-1000	Worker A	Sarah	Accept	17/03/2009 15:00:14	1 day

House cleaning service

Date	Time	Resource	Client	Status	Record time	Time-out
27/03/2010	1300-1500	Worker B	Edith	Accept	17/03/2009 11:40:20	3 days

Table 7.13: The example of diary slots after time-out

The diary's table includes the fields of 'Record time' and 'Time-out' to deal with the time-out mechanism of a DAS's diary. Before recording new data into the diary, the DAS will check whether the diary's record of the date and time specified by the new record has expired. The DAS compares the system's time with the data in the 'Time-out' field of the specified record. If the specified record in the diary is expired or time-out, then the DAS will update specified record with the new data. Otherwise, there will not be any operation for this specified record..

Time-out is involved in the DASs' decision making in dealing with the client's request and managing DASs' diary or resources. The time-out must be defined in the diary in advance. The record will be read to check time-out everytime. If the record has timed-out, it will be written with the new data.

7.2.7 Proposition and Selection Offers of the DASs and the P-Broker

Case

The consultant wants Edith the progress of her physiotherapy treatment to be assessed by the senior physiotherapist at the community hospital. Hence, she needs ambulance transport that collects her at the earliest time she can manage. The ambulance transport services are provided by several transport providers offering different kinds of ambulance vehicles and prices. The nurse modifies her care plan and makes a request to the providers to propose their offers about the ambulance transport services.

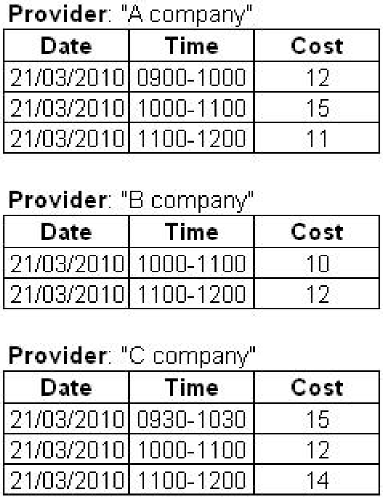
Operation

The nurse uses the CAPTAIN system to modify the integrated care plan for Edith as in the following.

1. Retrieve Edith's care plan from the system database
2. Add the ambulance transport service into the care plan and change its service detail according to her needs
3. Issue the requests to the P-Broker
4. Receive and select the providers' offers that meet Edith's needs
5. Accept and save the care plan and actual plan to the system's database

Result

Table 7.14 shows a number of offers for the ambulance transport service that are proposed by the different providers.



Provider: "A company"

Date	Time	Cost
21/03/2010	0900-1000	12
21/03/2010	1000-1100	15
21/03/2010	1100-1200	11

Provider: "B company"

Date	Time	Cost
21/03/2010	1000-1100	10
21/03/2010	1100-1200	12

Provider: "C company"

Date	Time	Cost
21/03/2010	0930-1030	15
21/03/2010	1000-1100	12
21/03/2010	1100-1200	14

Table 7.14: The example of DASs' offers

Each DAS representing a provider receives the requests from the P-Broker via the I-Broker. It translates the requests into local query formats and retrieves the relevant data from the data sources. Consequently, it employs the decision model to generate the offers for the client. For the purpose of this study, the DAS employs three main criteria when generating its offers.

- The cost preferred by the client
- The dates and times available for providing the services that are requested by the client
- The time-out period for the client's reservation.

The DAS's decision model will generate the three following types of offers for the client. Then the DAS will send them to the P-Broker via the I-Broker.

- The offer with the cheapest cost
- The offer with the earliest time
- The offer with the latest time

When the P-Broker receives the offers from the DASs, it uses the decision model to choose between the offers for the client. The decision model of the P-Broker employs two types of strategy to select the offers

- The P-Broker strategy

  The default or private strategy of the P-Broker is used by the decision model to select between the DASs' offers.

- The client strategy

  The client can define the strategy to be used with the expected result from the P-Broker. There are four types of client strategy, as follows.

  - 'Earliest time'
  - 'Latest time'
  - 'Cheapest cost'
  - 'I decide'.

Therefore, the P-Broker selects the offers for the client by following the sequence.

- Select three offers that provide the earliest and latest time together with the cheapest cost.
- Choose those selected offers based on the client's strategy.

For the case of Edith, the P-Broker chooses the offer of 'A Company' that it costs 12 pounds and provides service on 21/03/2010 during 0900-1000. If the client's strategy is 'I decide', the P-Broker will choose and return all of offers to the client so that the client can choose one of them.

From the perspective of service negotiation in CAPTAIN, the decision models employed by the P-Broker and the DASs are the main negotiation elements that are included in the negotiation model for CAPTAIN. The decision model together with the protocol for the negotiation are the key parts of the service negotiation that is conducted between the client and provider sides.

The P-Broker and the DASs represent the service requester and service providers. Both sides use their own decision models for decision-making dealing with various negotiation circumstances or issues in order to meet their objectives of service negotiation. The DASs have their own decision models that are used to generate the offers. The client and the P-Broker also have their own decision models or strategies that are used to response the DASs' offers. Depend on the purpose of the negotiation, different system components apply different strategies to negotiate with their participants.

7.2.8 System Observation

This section describes how the system behaviour of CAPTAIN is observed in terms of the main system components and the events that occur in the system during service negotiation in CAPTAIN. The aim is to make sure that the system components are performing their tasks properly. Firstly, it begins with the background of some approaches especially the partial ordering of the events. Then it investigates the ordering of events in service negotiation for CAPTAIN.

7.2.8.1 Ordering of Events in Distributed Systems

In a distributed system, ordering of events or processes is one of the key aspects that is needed by an observer in order to analyse the system for such purposes such as performance analysis, monitoring and debugging the system (Fidge, 1996; Raynal and Singhal, 1996). “Messages must be ordered according to their cause-and-effect relation to ensure correct behavior of the system” (Zuberi and Shin, 1996).

A partial ordering of the events is one of the popular approaches that is used to examine the ordering of events in a distributed system (Lamport, 1978). A distributed system consists of a group of different processes which are spatially separated. Each process consists of a sequence of events. Based on an application, the events in the process are mainly classified into three types.

- The exchange information by sending or receiving a message in the process
- The execution of a subprogram
- A single machine instruction on the computer

For any single process, the sequence of the events with a priori total ordering describes how an event takes place before or after the other event. In a particular sequence, one of two events occurs first. This partial ordering is defined as the “*happened before*” relation, denoted by “→”. The relation on a group of events of a system is regulated by the following three conditions.

- if *a* and *b* are events in the same process and *a* occurs before *b* then $a \rightarrow b$.

- If a is the sending of a message by one process and

b is the receipt of the same message by another process,

then $a \rightarrow b$.

- If $a \rightarrow b$ and $b \rightarrow c$ then $a \rightarrow c$.

(Lamport, 1978)

Therefore, this approach is useful for understanding or analysing any multi-process system. The events or discrete actions provide the means for an observer to observe what is occurring in the system (Fidge, 1996).

7.2.8.2 Ordering of Events in Service Negotiation for CAPTAIN

To investigate the ordering of events in service negotiation for CAPTAIN, an *event logger* is used, where records this also the timestamp of a sequence of the events, occurring when there are the execution of a subprogram or interactions among system components of CAPTAIN. For the purpose of this research, the event logger records the timestamp when the 'receiver' system component receives the message sent from the 'sender' system component that the event of the 'sender' "happened before" the one of the 'receiver'.

The event logger is employed as a web service. It receives event data from those system components during the process of service negotiation in CAPTAIN, and records the event data into a log file. There are two types of event data.

- The names of the system component
- The event together with its timestamp, the time at which a certain event occurred

The example of the event data is *event('P-Broker', 'receive offers', 'I-Broker', '2009-03-15 09:47:13.908').* This event occures in the P-Broker when it receives a message from the I-Broker on date and time as 2009-03-15 and 09:47:13.908 respectively. The second is set to three fractional digits that produces timestamps to millisecond precision.

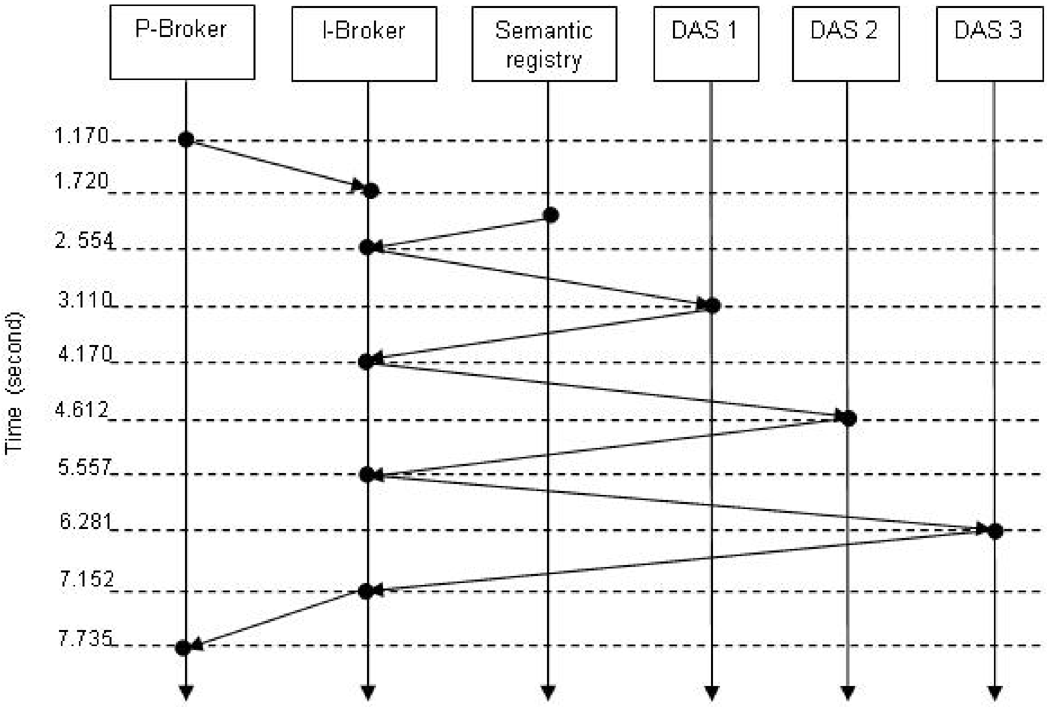


Figure 7.1: The ordering of events in service negotiation for CAPTAIN

Figure 7.1 shows the sequence diagram for the ordering of events in a service negotiation for CAPTAIN. It also illustrates the timestamps of the events of the system components interacting each other during service negotiation. The ordering of events starts with the P-Broker submitting a query to the I-Broker. Then, the "happened before" system components send messages to the receipt system components at a particular time and a specific point of the each event as shown in 7.1. Finally, the P-Broker receives the result from the I-Broker. These events form the serial events in timestamp order. Therefore, all system components process their tasks, and then, send and receive messages among them as in orders specified by CAPTAIN.

From the perspective of service negotiation in CAPTAIN, the ordering of the events implies that all system components are regulated by the negotiation process and protocol to work together to produce the integrated care plan for the client. The negotiation objects or messages are exchanged by sending and receiving among the system components, following the process of service negotiation serial.

7.3 Summary

This chapter presents the evaluation and reviews the 'proof of concept' results from using CAPTAIN. The evaluation has been based on a use case and its scenarios. They focuse on the three main features of the CAPTAIN application: function, data access and negotiation. The client can use and operate the application by using three main main service functions to produce the integrated care plan: create, modify and delete. For the data access, the providers provide data access services that are based on the 'read-write' property. During and after negotiation, the providers' data sources are accessed to read and write the negotiation content in their data sources. The CAPTAIN broker can negotiation with the providers to meet the purposes of the negotiation participants. The result of the negotiation is an integrated care plan that meets the needs of the client. All system components of CAPTAIN perform properly in order to work together to produce the integrated care plan for the client.

Chapter 8

Discussion

This chapter interprets and discusses the evaluation of CAPTAIN as presented in Chapter 7. It discusses the achievements and limitations of CAPTAIN, particularly as related to the system architecture, implementation and the negotiation model. Then, it examines how well CAPTAIN answers the research question.

8.1 Achievements

The evaluation, as presented in Chapter 7, focuses on the three distinctive features of the CAPTAIN application: the functions of the CAPTAIN broker, the role of data access and the negotiation model. The achievements of the CAPTAIN application are described as follows.

The functions of the CAPTAIN application

The CAPTAIN application performs the main service functions that meet the purpose of this research. The client can use and operate the application, through the CAPTAIN broker, by using three main service functions to produce and manage the integrated care plan: ‘Create’, ‘Modify’, and ’Delete’. The result of the integrated care plan meets the needs of a patient.

- Create

  The client can use the care plan template, as a pre-set care package, to create a new care plan for a patient. Each care plan template is specified in terms of the set of services needed for a given form of health and social care episode. The client chooses the care plan template that is suited for the patient. Then the clinician changes the details of the needs, preferences and constraints in the template so that they are appropriate for the patient. The changed template is finally issued to the system, through the P-Broker, to create a specific plan for the patient.

- Modify

  Modification to the integrated care plan occurs when the patient's needs change and deviate from the existing care plan. The client can modify the care plan and submit a new request to the P-Broker to negotiate with the DASs for new offers. The modification operation will also require some further service negotiation between the P-Broker and the DASs in order to agree the new terms and conditions.

- Delete

  An integrated care plan is deleted when the patient's case is closed. The integrated care plan is deleted from the domain knowledge. Any outstanding appointments in the data sources of the DASs are cancelled.

During service negotiation, some conflict may arise that requires the client to revise the integrated care plan. For the purpose of this research, the client has to cancel the less important service appointment and to make a new appointment through the DASs. Then the client makes the appointment for the new service at the specific date and time.

Data access

For the access services provided by the DASs, the active element of the role of data access (read-write) is the important new element provided for the service negotiation mechanism, especially for the DASs. The P-Broker can provide care planning services by negotiating with the DASs to create the integrated care plan for the patient. The negotiation between the P-Broker and the DASs is based on content. During and after negotiation, they exchange their negotiation objects through the I-Broker providing information services to the P-Broker. According to a request from the P-Broker, the I-Broker invokes the DASs' data access services to read and write the negotiation content in their data sources. The DASs evaluate the P-Broker's request, generate their offers and update the content based on the agreements between negotiation participants. This represents a substantial development from the read-only role performed by the DASs used with IBHIS.

Negotiation model

The negotiation model is the key element of the CAPTAIN application, needed if it is to perform properly. It consists of four main elements: negotiation process, negotiation protocol, negotiation object and decision model. From the evaluation in Chapter 7, we have been able to show that all negotiation elements of the negotiation model perform as expected. The main system components of CAPTAIN, including the client, follow the negotiation process and protocol by organising the exchange of negotiation objects among them before, during and after service negotiation in order to produce the integrated care plan for the client. Both of the negotiation participants, the P-Broker and the DASs, employ their own decision models that may involve several decision-making processes, for instance the selection of the DASs' offers by the P-Broker; and the evaluation of the P-Broker's requests and generation of the offers by the DASs. Each negotiation participant has its own strategy for dealing with the negotiation opponent. From the service requestor point of view, both the client and the P-Broker are involved in the decision-making of the selection of the DASs' offers. This operation is part of the decision model employed by the negotiation participants.

The system behaviour of the CAPTAIN system

The observation of the system behaviour of the CAPTAIN system shows that the main system components can perform their tasks and interact with each other correctly during service negotiation. The system components process the tasks and exchange their messages by sending and receiving the messages among them as in the sequence of events or the processes of service negotiation. The sequence of the events with a priori total ordering illustrates that an event takes place before or after the other event. Therefore, the proper ordering of the events entails that the main system components of the CAPTAIN system are regulated to work together in order to produce the integrated care plan that meets the needs of the client.

The main characteristics of the CAPTAIN application

From the implementation aspect, the CAPTAIN application has been developed and implemented by using 'state of art' service technologies. The application performs as a proof of concept service-oriented broker application that deals with heterogenous, distributed and autonomous data sources. It is a flexible and adaptable planning system demonstrating a full range of service-based software characteristics: description, discovery, negotiation, delivery and composition. All system components work together properly to produce an integrated care plan for the client. Particular characteristics of the application include:

1. *Adaptable plan*

   The CAPTAIN application provides planning services that produce an adaptable integrated care plan that is based on the needs of the client. The client can revise and renegotiate the integrated care plan whenever the client's needs change.

2. *Resource flexibility*

   The DASs provide for resource flexibility by managing their resources diaries in response to the queries of the P-Broker. The resources, that are available to provide services to the patient, consist of service appointments in the DASs' diaries and identified in the patient's integrated care plan.

3. *Dynamic data integration*

   The P-Broker of the CAPTAIN application dynamically integrates data on demand, through the I-Broker and semantic registry, where the data is drawn from various DASs that are autonomous, heterogeneous and distributed data sources.

4. *Platform independent*

   The CAPTAIN application is platform independent. It is capable of being installed and run on different kinds of standard platforms, such as Microsoft Windows and Linux. Its web user interface also allows the client to interact with it through any internet browser.

8.2 Limitations

The purpose of this research has been to investigate and develop the CAPTAIN system, a service-oriented broker application that includes negotiation elements to perform service negotiation tasks among the system components. However, there are limitations of the "proof of concept" demonstration system in terms of some of its aspects, which are described as follows.

Semi-automated negotiation system

CAPTAIN is a semi-automated negotiation system that, through the P-Broker, involves the client in negotiation with the DASs. The client defines the strategy for obtaining an expected result to the client's request, so that the P-Broker employs the client's strategy together with its own strategy to negotiate with the DASs and then to select from the offers made by the DASs. Then the P-Broker presents the selected offers so that the client chooses them and then acknowledges the final chosen offers to the DASs. It is therefore necessary to involve the client in much of the decision-making involved.

Reverse auction mechanism

The approach used for service negotiation in CAPTAIN is based on a reverse auction mechanism, whereby the P-Broker, acting on behalf of the client, solicits offers about services from a set of potential DASs. However, the reverse auction mechanism has some limitations, which includes.

- The decision model of the P-Broker uses a fixed strategy. When the P-Broker receives offers from the DASs, it selects offers with the 'earliest time', 'latest time' or 'cheapest cost' according to the client's requirements. Hence the decision model is not flexible enough to be tailored to meet more complicated terms and conditions of a client's needs, in response to the providers' constraints.

- Only the P-Broker can propose a request to the DASs. The auction mechanism does not allow the P-Broker and the DASs to negotiate with each other by exchanging their offers and making counter offers until both negotiation participants are satisfied with the results. Therefore, in such circumstances, the P-Broker's strategy cannot be adapted to respond to specific offers from the DASs.

Consequently, while the reverse auction mechanism demonstrates the potential for using negotiation, the resulting integrated care plan produced does not necessarily fit the best offers for the patient.

Domain knowledge

From the system implementation point of view, the CAPTAIN's domain knowledge retains only the care plan and actual care plan together with the patient records and system service priority. The domain knowledge does not hold and provide data and information related to the service negotiation that can be used by the P-Broker to negotiate with the DASs, such as a negotiation policy and strategy. Such a negotiation policy and strategy would be useful to enable the P-Broker to negotiate with the DASs to obtain more appropriate service offers for the client.

Synchronous interaction

In terms of system performance, the interaction or communication between the I-Broker and the DASs during negotiation is synchronous. With this method, the I-Broker sends its request to the DAS and waits until it receives a response from the DAS. Consequently, the use of a synchronous mechanism can cause considerable delay, especially when the I-Broker interacts with many DASs. To overcome this problem, the communication mechanism for the CAPTAIN system should really be asynchronous. This would enable the I-Broker to invoke the DASs' data services without waiting until a DAS has responded back. In the meantime, the I-Broker could also invoke the other DASs' data services. Adapting this form could improve the performance of the CAPTAIN system.

To improve the negotiation process, the CAPTAIN system should include the care planning management service into the P-Broker in order to deal with the offers from the DASs. To do this, however, CAPTAIN would need to adapt its concepts and system in some aspects, especially the system architecture, the negotiation model, and the system implementation as discussed below.

1. The negotiation protocol should be based on a bargaining model rather than an auction. The P-Broker and the DASs can then exchange their offers and counter-offers during the negotiation until both sides achieve the mutual agreements on terms and conditions.

2. The negotiation model or the decision model should include some negotiation elements that are related to the decision-making process of the P-Broker, for instance policy and strategy. The negotiation policy is "A general guiding principle for achieving a business negotiation goal under some specified conditions", while the negotiation strategy is defined as "The specifications of decisions and actions for implementing negotiation policies" (Li *et al.*, 1998). One option is for the P-Broker to be able to alter its strategies on the fly to negotiate with the DASs and to manage the care plan by evaluating the offers against the care plan.

3. The domain knowledge includes not only the patient, template, care and actual plans but also the information related to service negotiation, such as policy, strategy, tactics, feedback from negotiation and the other kinds of decisions model, for instance a cost-benefit model. Therefore, the care planning service can use this information to negotiate with the DASs and to manage the care plan to meet the needs of the patient.

The scenario "Management of Appointment Conflict" in Section 7.2.3 is an example of how the integrated care plan can potentially be improved by the care planning management service that includes the negotiation policy and strategy. By using the policy and strategy, the care planner can define various kinds of policies and strategies that are flexible for the P-Broker to deal with different negotiation circumstances. The house task and physiotherapy services have made appointments for Edith on Tuesday and Thursday afternoons respectively. However, a district nurse needs to make an appointment and the nurse will be only free at the same time as the appointed times of the house task and physiotherapy. The P-Broker may solve the conflict of appointment by employing the negotiation policy and strategy to rearrange the best appointments for these services.

The other limitation is the multiple service requests that the P-Broker formulates into the sequence of services. To deal with the complex multiple requests, it may need to use a decision making model, such as Constraint Satisfaction Problems (CSP) in order for the P-Broker to handle more complex requests.

8.3 Addressing the Research Question

This research has investigated and developed a service-oriented broker application that incorporates elements of negotiation in order to demonstrate a full range of service characteristics. It was motivated by the limitations of the previous demonstration of a service based distributed information management system, where sources acted in a passive role, providing information to the client. The IBHIS service-based broker architecture addressed only information gathering, with the role of service negotiation being limited to access control. These limitations led to the research question:

> "*Can we extend the service-based broker architecture incorporating negotiation elements to enable the development of a planning bro-*

kering service system that is flexible?"

The concept of the planning broker is one means of addressing the research question. Its approach to the development of a planning system is one of extending the service-based broker architecture. The main contribution to accomplishing the extended architecture is by adding the planning service broker, together with modifying the role of service negotiation and the DASs. The extended architecture includes the planning, negotiation and software service-based models so that the planning system is flexible and adaptable enough to deal with the dynamic needs of the client and the constraints of the providers. The three models involved define the characteristics of the system as follows.

- The planning model is the central structure of the planning system for organising resources to meet the needs of the client.
- The software service-based model provides for decoupling from individual resources. It provides the dynamic service context that can find and use new resources as they become available, without a need to use a fixed set of permanent resources.
- The negotiation model enables adaptation of the details of the plan, based on the interest of the client, by interacting with the providers.

Scalability is a major outcome of the extension of the software service-oriented broker architecture. The service-oriented broker software can be developed and extended by including a new service broker, through a service supply chain within a software services open market. "Services are supplied through a service supply chain" from primitive services to consumers (Bennett *et al.*, 2000). The primitive services, at the bottom of the chain, are the information broker services providing information gathered from diverse data services of distributed, heterogenous and autonomous data sources. The information broker services provide their information to a group of various types of service brokers, at the higher layer of the service supply chain. These service brokers are formed as a hierarchy of software services providers: supplier-supplier (wholesale) and consumer-supplier market (retail). Each service broker itself can be either the software service supplier or consumer, or both. The service broker may compose a number of lower layer broker services to provide its services to the consumer. The final service provision of these broker software services, through the service supply chain, is to the consumers, who are at the top of the chain.

The new service broker then can be added to the service supply chain. The software service-oriented broker architecture can be potentially extended with the new service broker. Therefore, the architecture forms the the basis for incorporating different kinds of service-oriented broker application with this structure.

The other main outcome of this project is the flexibility of service interaction between the service brokers that is made possible by the negotiation model. Each negotiation element has a different role for accommodating the flexible interaction among the brokers.

1. The negotiation process is a dynamic end-to-end process starting from the service request and ending up with a resolution. The negotiation process consists of three main stages as follows.

   (a) The pre-negotiation stage accommodates the broker to prepare and produce the initial information for the service negotiation stage. The intial information includes a service description published by the broker.

   (b) For the service negotiation stage, the broker negotiates with the providers to obtain the best deal results for the client.

(c) The post-negotiation stage deals with the negotiation results from the service negotiation stage. The client is involved in the decision making about the results. The providers may be acknowledged with the result of the client's decision making.

2. The negotiation protocol defines the negotiation interaction mechanism following any agreed rules by the brokers.

3. The negotiation object is used by the service brokers to communicate and exchange information with each other during service negotiation or interaction.

4. The decision model consists of the rules employed by the service brokers for their decision making. The brokers have their own decision models which are internal and private decision processes. Each broker has its own strategy for interacting or dealing with the other brokers' response or actions, so that the broker can achieve its negotiation objective.

Therefore, the negotiation model should allow the flexible tailoring of a number of other candidate brokers' services so that the broker can obtain the required services that offers the best deal from those candidate brokers.

8.4 Summary

We have interpreted and discussed the evaluation of CAPTAIN. We have also discussed the achievements and limitations of CAPTAIN, particulary the system architecture and implementation as well as negotiation model. Finally we discussed how well CAPTAIN answers the research question.

The demonstration of the CAPTAIN application, as the 'proof of concept', shows that it achieves the main objectives of this thesis focusing on three main features: function, service negotiation and data access (read-write). The CAPTAIN application provides a number of major characteristics of service-oriented software, especially adaptable plan, resource flexibility, dynamic data integration and platform independent. However, there are some limitations of the CAPTAIN application, for instance its use of semi-automated negotiation and synchronous interaction. To answer the research question, the concept of the planning broker is one approach to extending the service-based broker architecture in order to enable the development of a planning brokering system. The resulting planning system is flexible and adaptable because of the service-oriented broker architecture that includes the planning, negotiation and software service-based models. The primary outcomes of this thesis are to demonstrate the scalability of the software service-oriented broker architecture and the flexibility of service interaction between the planning service brokers.

Chapter 9

Conclusions

This chapter presents the conclusion of the thesis, summarizes its contribution as original work and identifies directions for further work.

9.1 Summary

The aim of this thesis has been to develop and investigate a service-oriented broker architecture that includes an element of service negotiation. The research method adopted for this study was based on “Concept Implementation”. The objective was to develop a “proof of concept” system that could demonstrate the feasibility of the proposed model. A service-based broker architecture was therefore developed and evaluated through a case study. The key issues and problems of the case study were defined within a particular context of the healthcare sector. A negotiation model was also developed for the service negotiation and included in the architecture. The key features of the resulting prototype were demonstrated by using a set of scenarios based on the use case, and used to evaluate the potential of the extended service-based broker architecture.

This thesis has therefore explored the concepts and role of a broker employed in a service-oriented application that includes negotiation elements in order to demonstrate a full range of service characteristics: description, discovery, negotiation, delivery and composition. It is called the CAPTAIN broker (Care Planning Through Auction-based Information Negotiation). The service-oriented broker architecture of CAPTAIN has extended the concepts and role of the broker model, service negotiation and data access used in IBHIS (Integration Broker for Heterogeneous Information Sources). IBHIS is based on the concept of a software service-based model, called Software as a Service (SaaS). The IBHIS information broker performs by providing information services that gather data from distributed, heterogeneous and autonoumous data sources.

CAPTAIN includes planning, negotiation and service-based models to provide a flexible and adaptable planning system. It uses an integrated care plan to provide a research case study. The CAPTAIN application has been developed based on three main foundations: software service-based models, brokering services and a service supply chain. It produces an adaptable plan and provides flexible resources in response to the needs of a client by dynamically integrating data from diverse data sources that are autonomous, heterogeneous and distributed. In addition, it is platform independent and so able to be implemented and executed on various types of standard platforms, for instance Microsoft Windows.

The CAPTAIN application provides a planning broker service for a client. In performing the role of a broker, the CAPTAIN broker acts on behalf of the client to negotiate with the service providers in order to reach agreement of terms and conditions of services supplied. The negotiation process uses a reverse auction model to address the conflict of interests between the client and the service providers. The result of the negotiation should then be an integrated care plan that meets the needs of the client.

The prototype proof of concept implementation was based on Java 2 Enterprise Edition (J2EE) Web Services technologies. It has been developed and run within the Eclipse Java EE IDE for Web Developers environment working on a Windows platform. It supports current Web services standards, such as SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description, Discovery and Integration) and XML (Extensible Markup Language). The CAPTAIN application is a dynamic web application. All main system components of CAPTAIN, except the semantic registry, are deployed as Web services.

The demonstration of the CAPTAIN application has been based on a use case and its scenarios. The evaluation of the CAPTAIN application has addressed the three main features: function, data access and negotiation. The result of the evaluation has shown that the CAPTAIN system meets the aim or the criteria for success of this study as follows.

1. The functions of the CAPTAIN broker

   The CAPTAIN application can perform by providing a care planning service through the CAPTAIN planning broker to a client. The client can use and operate the application to produce and manage an integrated care plan by using the application's three main service functions: create, modify and delete. The result in the form of the integrated care plan can then meet the needs of a patient.

2. The role of data access (read-write)

   The CAPTAIN broker performs the role of a care planning service broker while the service providers or the DASs perform an active role to negotiate with the planning broker in order to achieve the objective of an client. They negotiate via an information broker providing information services to the planning broker. The service negotiation is based on content. If the planning broker and the providers agree on terms and conditions of the providers' service provision, the information broker can access the providers' data access services with the 'read-write' property in order to update the content in their data sources with the new status of the service negotiation.

3. The negotiation model for CAPTAIN

   The negotiation model is the key part of the service negotiation. It involves all of the main system components in order to produce the integrated care plan to meet the agreement between the CAPTAIN broker and the service providers. The negotiation model also permits the plan to be adapted, based on the client's interest, by interaction with the providers.

In addition, the CAPTAIN system is a distributed system consisting of various separated system components. CAPTAIN has demonstrated that the main system components can properly perform their tasks and interact each other by exchanging their negotiation objects among them as in the ordering of events regulated by CAPTAIN. Therefore, the proper ordering of the event should lead to the system's performance producing an appropriate result that meets the objective of the client.

The two primary outcomes of this project are as follows.

1. Scalability of the software service-based broker architecture

   The separation of concerns, demonstrated by the combination of planning and information brokers within the software service-oriented broker architecture of CAPTAIN, is potentially capable of being scaled into a larger group of different kinds of service brokers. A new broker service can extend the service-oriented broker architecture, through the service supply chain within a software services open market in which the primitive information services are provided by information brokers. Each broker can act as a service provider or a third party composing and tailoring fine-grained services from other service brokers in order to obtain the best deal for a consumer. However, the consumer itself may be the service provider supplying its services to the other consumer or service broker. Therefore, the extended service-oriented broker architecture can be used as the basis for various types of service-oriented broker applications for different purposes.

2. Flexibility of service interaction between the planning service brokers

   The negotiation model makes it possible to employ and apply flexible interactions among the service brokers. The model allows tailoring of a number of other brokers' services in order to obtain the plan that is offered the best deal from a selected broker or third party.

An example of the outcome is travel services. A packaging travel broker can provide a service of planning of a journey package including flight and hotel plans for a consumer. The packaging travel broker would find both flight and hotel planning brokers from the other brokers in the travelling software services market, and negotiate with them for the flight and hotel plans that each offers the best deal.

When there is a further broker service for rental cars, the new rental car broker is added into the the travelling software services market. Therefore, if the consumer wants a plan for the rental cars, the packaging travel broker can also find the rental car broker and negotiate with them for tailoring the best plan from a group of rental car providers.

In conclusion, the concept of the service-oriented broker architecture can be extended to enable a scalable and flexible distributed information management with interaction. A new service broker can be included into the architecture so that it can perform diverse types of distributed information management services. In addition, the service broker can interact with the other service brokers by employing the negotiation model to deal with the conflicts of interest among them. Therefore the extended distributed information management system with interaction is scalable and flexible.

9.2 Further Work

Based on the research in this thesis, there are some questions left unanswered by the limitations of this research. These lead to a number of possible areas for further work as follows:

1. The negotiation model

   The capability of the broker could be enhanced to deal with more complicated negotiation scenarios. This would require the negotiation model to be extended by including new negotiation elements, such as a negotiation policy and a negotiation strategy. The broker would then be able to perform more flexible service negotiation by choosing appropriate strategies and policies for negotiating with the service providers.

2. The negotiation mechanism

   The negotiation protocol that is based on an auction mechanism is not sufficiently flexible to handle negotiation involving complicated terms and conditions that reflect continous and rapid changes of a client's needs against the providers' constraints. Therefore the service-oriented broker application needs to adopt a bargaining model to deal with such complicated negotiation situation. The bargaining mechanism can allow negotiation participants to negotiate with each other by exchanging offers and counter offers until both sides are satisfied with the results. Thus the use of a bargaining mechanism should lead to better agreement for both sides.

3. The role of domain knowledge

   The role of domain knowledge could be extended to retain and provide data and information related to the service negotiation, for instance a negotiation policy and strategy. Hence the broker could keep relevant negotiation information and uses it to guide future negotiation with the providers.

4. Automated negotiation based on a semantic-based framework

   A semantic-based framework could be employed to support automated negotiation within the service-oriented broker architecture so that the negotiation can be more adaptable and flexible, for instance a semantic-based framework or ontologies for supporting negotiation in service oriented architectures (Comuzzi *et al.*, 2009) and electronic commerce (Tamma *et al.*, 2005). Hence, the service-oriented broker application should support a wide range of negotiation mechanisms.

5. The concepts and role of the service broker

   The concepts and role of the service broker should be applied to other aspects of service provision for a client, for instance electric commerce that has been dominant in the global software services market for decades. Consequently, the service-oriented broker architecture should be modified from the planning service broker to the other type of the service brokers, such as a buyer service broker. The result of this further work should demonstrate that the service-oriented broker architecture can be used not only for the planning within healthcare context but also in other service contexts.

Bibliography

Ausubel, L. M., Cramton, P. and Deneckere, R. J. (2002), Bargaining with incomplete information, Technical report, University of Maryland, Department of Economics.

Bajari, P., McMillan, R. and Tadelis, S. (2009), 'Auctions versus negotiations in procurement: An empirical analysis', *The Journal of Law, Economics, & Organization* **25**(2), 372–399.

Barghouti, N. S., Naser, C., Barghouti, S., Kaiser, G. E. and Kaiser, G. E. (1994), 'Concurrency control in advanced database applications', *ACM Computing Surveys* **23**, 269–317.

Beam, C. and Segev, A. (1996), 'Electronic catalogs and negotiations', *CITM Working Paper 96-WP-1016* . Fisher Center for Information Technology & Management, University of California, Berkeley, US.

Benatallah, B., Casati, F., Toumani, F. and Hamadi, R. (2003), Conceptual modeling of web service conversations, *in* 'Proceedings of the 15th International Conference on Advanced Information Systems Engineering', Springer-Verlag, Berlin, Heidelberg, pp. 449–467.

Bennett, K., Gold, N., Layzell, P., Zhu, F., Brereton, P., Budgen, D., Keane, J., Kotsiopoulos, I., Turner, M., Xu, J., Almilaji, O., Chen, J. and Owrak, A. (2003), 'A broker architecture for integrating data using a web services environment', *Service Oriented Computing* **2910**, 409–422.

Bennett, K. H. and Xu, J. (2003), Software services and software maintenance, *in* 'Proceedings of the Seventh European Conference on Software Maintenance and Reengineering', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, p. 3.

Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L. and Munro, M. (2000), Service-based software: the future for flexible software, *in* 'Proceedings of the Seventh Asia-Pacific Software Engeering Conference (APSEC)', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, pp. 214 – 221.

Bennett, K., Munro, M., Gold, N., Layzell, P., Budgen, D. and Brereton, P. (2001), An architectural model for service-based software with ultra rapid evolution, *in* 'Proceedings of the IEEE International Conference on Software Maintenance (ICSM)', Vol. 0, IEEE Computer Society, p. 292.

Bianchini, D., De Antonellis, V., Pernici, B. and Plebani, P. (2006), 'Ontology-based methodology for e-service discovery', *Information Systems* **31**(4), 361–380.

Bichler, M. (2000*a*), 'An experimental analysis of multi-attribute auctions', *Decision Support System* **29**(3), 249–268.

Bichler, M. (2000*b*), A roadmap to auction-based negotiation protocols for electronic commerce, *in* 'Proceedings of the 33rd Hawaii International Conference on System Sciences', Vol. 6, IEEE Computer Society, Washington, DC, USA, p. 6018.

Bichler, M. and Kalagnanam, J. (2005), 'Configurable offers and winner determination in multi-attribute auctions', *European Journal of Operational Research* **160**(2), 380–394.

Bichler, M., Kaukal, M. and Segev, A. (1999), ‘Multi-attribute auctions for electronic procurement’, *IBM IAC Workshop on Internet Based Negotiation Technologies* . IBM T.J. Watson Research Center, USA.

Bichler, M., Kersten, G. and Strecker, S. (2003), ‘Towards a structured design of electronic negotiations’, *Group Decision and Negotiation* **12**, 311–335.

Bichler, M. and Segev, A. (2001), ‘Methodologies for the design of negotiation protocols on e-markets’, *Computer Networks: The International Journal of Computer and Telecommunications Networking* **37**(2), 137–152.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C. and Orchard, D. (2004), ‘Web services architecture’. W3C Working Group Note, World Wide Web Consortium (W3C), [Online] Available from: http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/. [Accessed: 1st August 2010].

Brereton, P. and Budgen, D. (2000), ‘Component-based systems: A classification of issues’, *Computer* **33**(11), 54–62.

Budgen, D., Brereton, P. and Turner, M. (2004), Codifying a service architectural style, *in* ‘Proceedings of the 28th Annual International Computer Software and Applications Conference’, Vol. 1, IEEE Computer Society, pp. 16–22.

Budgen, D., Rigby, M., Brereton, P. and Turner, M. (2007), ‘A data integration broker for healthcare systems’, *Computer* **40**(4), 34–41.

Budgen, D., Turner, M., Kotsiopoulos, I., Zhu, F., Russell, M., Rigby, M., Bennett, K., Brereton, P., Keane, J. and P. (2005), Managing healthcare information: The role of the broker, *in* ‘Proceedings of Healthgrid’, Vol. 112, IOS Press, pp. 3–16.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J. and Brandic, I. (2009), ‘Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility', *Future Generation Computer Systems* **25**(6), 599–616.

Canfora, G. and Penta, M. (2009), Service-oriented architectures testing: A survey, Springer-Verlag, Berlin, Heidelberg, pp. 78–105.

Carles, P. F., Sierra, C. and Jennings, N. R. (1998), 'Negotiation decision functions for autonomous agents', *International Journal of Robotics and Autonomous Systems* **24**, 3–4.

Challis, D., Chessum, R., Chesterman, J., Luckett, R. and K., T. (1990), *Case Management in Social and Health Care*, Personal Social Services Research Unit, The University of Kent, Kent, UK.

Chen, E., Kersten, G., Neumann, D., Vahidov, R., Weinhardt, C. and Yu, B. (2007), A framework for e-market system assessment and design, *in* G. E. Kersten, J. Rios and E. Chen, eds, 'Group Decision and Negotiation (GDN)', Montreal, Canada.

Chen-Ritzo, C.-H., Harrison, T. P., Kwasnica, A. M. and Thomas, D. J. (2005), 'Better, faster, cheaper: An experimental analysis of a multiattribute reverse auction mechanism with restricted information feedback', *Management Science* **51**(12), 1753–1762.

Chung, J.-Y., Lin, K.-J. and Mathieu, R. G. (2003), 'Web services computing: advancing software interoperability', *Computer* **36**(10), 35–37.

Comuzzi, M., Kritikos, K. and Plebani, P. (2009), A semantic based framework for supporting negotiation in service oriented architectures, *in* 'Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing', IEEE Computer Society, Washington, DC, USA, pp. 137–145.

Comuzzi, M. and Pernici, B. (2005), An architecture for flexible web service qos negotiation, *in* 'Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference', IEEE Computer Society, Washington, DC, USA, pp. 70–82.

Comuzzi, M. and Pernici, B. (2009), 'A framework for qos-based web service contracting', *ACM Transaction on the Web* **3**(3), 1–52.

CQC (2010), 'The care quality commission'. Care Quality Commission.
URL: *http://www.cqc.org.uk/ [Accessed: 1st August 2010]*

Cusumano, M. A. (2008), 'The changing software business: Moving from products to services', *Computer* **41**, 20–27.

Dang, J. and Huhns, M. N. (2005), Coalition deal negotiation for services, *in* 'Proceedings of the Rational, Robust, and Secure Negotiation Mechanisms in Multi-Agent Systems (RRS) on Multi-Agent Systems', IEEE Computer Society, Washington, DC, USA, p. 67.

David, E., Azoulay-Schwartz, R. and Kraus, S. (2006), 'Bidding in sealed-bid and english multi-attribute auctions', *Decision Support System* **42**(2), 527–556.

Demirkan, H., Kauffman, R. J., Vayghan, J. A., Fill, H.-G., Karagiannis, D. and Maglio, P. P. (2008), 'Service-oriented technology and management: Perspectives on research and practice for the coming decade', *Electronic Commerce Research and Applications* **7**(4), 356–376.

Deventer, O., Tilanus, P., Schenk, M., Cramer, E. and Adriaanse, J. (2009), *Towards context information brokering*, A European Research Perspective, chapter Towards the Future Internet, pp. 250–262.

Devlin, B. (2006), 'Opening the door to a service oriented architecture'. IBM Workplace, IBM.

URL: *www.ibm.com/developerworks/websphere/tutorials/wes-helloport/resources.html [Accessed: 1st August 2010]*

Di Nitto, E., Ghezzi, C., Metzger, A., Papazoglou, M. and Pohl, K. (2008), 'A journey to highly dynamic, self-adaptive service-based applications', *Automated Software Engineering* **15**(3-4), 313–341.

DoH (2010), 'Department of health'. Department of Health, United Kingdom.
URL: *http://www.dh.gov.uk/ [Accessed: 1st August 2010]*

DoH-COI (2007), 'The dh guide: A guide to what we do and how we do it'. Central Office of Information, Depart of Health.
URL: *www.dh.gov.uk/en/Aboutus/HowDHworks/ [Accessed: 1st August 2010]*

Domingue, J., Fensel, D., Davies, J., González-Cabero, R. and Pedrinaci, C. (2009), *The Service Web: a Web of Billions of Services*, A European Research Perspective, chapter Towards the Future Internet, pp. 203–216.

Dunne, P. E., Wooldridge, M. and Laurence, M. (2003), 'The complexity of contract negotiation', *Artificial Intelligence* **164**, 2005.

EHMA (2010), 'Integrated for older people'. European Health Management Association, Belgium.
URL: *www.ehma.org/carmen/index.html/ [Accessed: 1st August 2010]*

Elfatatry, A. (2002), Service-oriented software: A negotiation process, PhD thesis, Institute of Science and Technology, University of Manchester, Manchester, UK.

Elfatatry, A. (2007), 'Dealing with change: components versus services', *Communications ACM* **50**(8), 35–40.

Elfatatry, A. and Layzell, P. (2004), 'Negotiating in service-oriented environments', *Communications ACM* **47**(8), 103–108.

Erl, T. (2007), *SOA: principles of service design*, Prentice Hall Press, Upper Saddle River, New Jersey, USA.

Faratin, P. (2000), Automated service negotiation between autonomous computational agents, PhD thesis, Queen Mary & West field College, London, UK.

Fatima, S. S., Wooldridge, M. and Jennings, N. R. (2004*a*), 'An agenda-based framework for multi-issue negotiation', *Artificial Intelligence* **152**(1), 1–45.

Fatima, S. S., Wooldridge, M. and Jennings, N. R. (2005), 'Bargaining with incomplete information', *Annals of Mathematics and Artificial Intelligence* **44**(3), 207–232.

Fatima, S., Wooldridge, M. and Jennings, N. R. (2004*b*), Optimal negotiation of multiple issues in incomplete information settings, *in* 'Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems', Vol. 3, IEEE Computer Society, Washington, DC, USA, pp. 1080–1087.
URL: *http://dx.doi.org/10.1109/AAMAS.2004.205*

Fidge, C. (1996), 'Fundamentals of distributed system observation', *IEEE Software* **13**(6), 77–83.

Fiorentini, X., Rachuri, S., Ray, S. and Sriram, R. D. (2009), Towards a method for harmonizing information standards, *in* 'Proceedings of the fifth annual IEEE International Conference on Automation Science and Engineering (CASE)', IEEE Press, Piscataway, New Jersey, USA, pp. 466–471.

Foster, I. T. (2005), Globus toolkit version 4: Software for service-oriented systems, *in* H. Jin, D. A. Reed and W. Jiang, eds, 'Proceedings of the fifth annual IEEE International Conference on Automation Science and Engineering', Vol. 3779 of *Lecture Notes in Computer Science*, Springer, pp. 2–13.

Gerding, E. H., Somefun, D. J. A. and Poutré, J. A. L. (2006), Multi-attribute bilateral bargaining in one-tomany setting, *in* 'Agent Mediated Electronic Commerce VI, Springer Lecture Notes in AI', Vol. 3435, Spriver-Verlag, pp. 129–142.

Gibbons, R. (1992), *A Primer in Game Theory*, Harvester Wheatsheaf, London.

Gimpel, H., Jennings, N. R., Kersten, G. E., Ockenfels, A. and Weinhardt, C. (2008), *Negotiation, Auctions, and Market Engineering*, Springer, Dagstuhl Castle, Germany.

Glass, R. L., Ramesh, V. and Vessey, I. (2004), 'An analysis of research in computing disciplines', *Communications of the ACM* **47**(6), 89–94.

Gold, N., Knight, C., Mohan, A. and Munro, M. (2004), 'Understanding service-oriented software', *IEEE Software* **21**(2), 71–77.

Gunasekaran, A. and Ngai, E. W. (2009), 'Modeling and analysis of build-to-order supply chains', *European Journal of Operational Research* **195**(2), 319–334.

Gunasekaran, A. and Ngai, E. W. T. (2004), 'Information systems in supply chain integration and management', *European Journal of Operational Research* **159**(2), 269–295.

Guttman, R. H. and Maes, P. (1998), Cooperative vs. competitive multi-agent negotiations in retail electronic commerce, *in* 'Proceedings of the Second International Workshop on Cooperative Information Agents (CIA)', Vol. 1435, SpringLink, pp. 135–147.

Heitlager, I., Jansen, S., Helms, R. and Brinkkemper, S. (2006), Understanding the dynamics of product software development using the concept of coevolution, *in* 'Second International IEEE Workshop on Software Evolvability', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, pp. 16–22.

Hung, P. C. K., Li, H. and Jeng, J.-J. (2004), Ws-negotiation: An overview of research issues, *in* 'Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS)', Vol. 1, IEEE Computer Society, Hawaii, USA.

IBM-CloundComputing (2010), 'Cloud computing'. IBM.
URL: *www.ibm.com/ibm/cloud/ [Accessed: 1st August 2010]*

IBM-SaaS (2010), 'Ibm: Software as a service (saas)'. IBM.
URL: *www-304.ibm.com/isv/marketing/saas/index.html [Accessed: 1st August 2010]*

IBM-SOA (2007), 'Smart soa: Best practices for agile innovation and optimisation.'. IBM.
URL: *www.findwhitepapers.com/whitepaper2340 [Accessed 1 August 2010]*

IBM-SOA (2010), 'Ibm: Service-oriented architecture (soa)'. IBM.

Irwin, D., Chase, J., Grit, L., Yumerefendi, A., Becker, D. and Yocum, K. G. (2006), Sharing networked resources with brokered leases, *in* 'Proceedings of the annual conference on USENIX '06 Annual Technical Conference', USENIX Association, Berkeley, CA, USA, pp. 18–18.

Jennings, N. R. (2000), Automated haggling: Building artificial negotiators, *in* 'Proceedings of the 6th Pacific Rim international conference on Artificial intelligence', Springer-Verlag, Berlin, Heidelberg, p. 1.
URL: *http://portal.acm.org/citation.cfm?id=1764967.1764969*

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Wooldridge, M. and Sierra, C. (2001), 'Automated negotiation: Prospects methods and challenges', *Group Decision and Negotiation* **10**(2), 199–215.

Jones, S. (2005), 'Toward an acceptable definition of service', *IEEE Software* **22**(3), 87–93.

Kersten, G. E., Michalowski, W., Szpakowicz, S. and Koperczak, Z. (1991), 'Restructurable representations of negotiation', *Management Science* **37**(10), 1269–1290.

Kerzner, H. (2003), *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, Wiley.

Kim, J. B. and Segev, A. (2003), A framework for dynamic ebusiness negotiation processes, *in* 'Proceedings of IEEE International Conference on Electronic Commerce (CEC', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, p. 84.

Kodner, D. L. and Spreeuwenberg, C. (2002), 'Integrated care: meaning, logic, applications, and implications - a discussion paper', International Journal of Integrated Care (IJIC).
URL: *www.ncbi.nlm.nih.gov/pmc/articles/PMC1480401/ [Accessed: 1st August 2010]*

Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009), 'Identification and analysis of business and software services—a consolidated approach', *IEEE Transactions on Service Computing* **2**(1), 50–64.

Kohlborn, T., Korthaus, A. and Rosemann, M. (2009), 'Business and software service lifecycle management', *Enterprise Distributed Object Computing Conference, IEEE International* **0**, 87–96.

Kontogiannis, K., Lewis, G. A., Smith, D. B., Litoiu, M., Muller, H., Schuster, S. and Stroulia, E. (2007), The landscape of service-oriented systems: A research perspective, *in* 'Proceedings of the International Workshop on Systems Development in SOA Environments', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, p. 1.

Krafzig, D., Banke, K. and Slama, D. (2004), *Enterprise SOA: Service Oriented-Architecture Best Practices*, Prentice Hall PTR.

Lamport, L. (1978), 'Time, clocks, and the ordering of events in a distributed system', *Communications of the ACM* **21**(7), 558–565.

Laplante, P. A., Zhang, J. and Voas, J. (2008), 'What's in a name? distinguishing between saas and soa', *IT Professional* **10**(3), 46–50.

Layzell, P. J. (2001), Implications of a service-oriented view of software, *in* 'Proceedings of the IFIP TC8/WG8.2 Working Conference on Realigning Research and Practice in Information Systems Development', Kluwer, B.V., Deventer, The Netherlands, The Netherlands, pp. 241–256.

Legner, C. and Vogel, T. (2007), 'Design principles for b2b services - an evaluation of two alternative service designs', *IEEE International Conference on Services Computing* **0**, 372–379.

Lewicki, R., Saunders, D. and Barry, B. (2006), *Negotiation 5th edition*, McGraw-Hill, Singapore.

Li, H. (2001), Automated E-business Negotiation: Model, Life Cycle and System Architecture, PhD thesis, Department of Computer and Information Science and Engineering, University of Florida, Florida, USA.

Litoiu, M., Mihaescu, M., Ionescu, D. and Solomon, B. (2008), Scalable adaptive web services, *in* 'Proceedings of the second international workshop on Systems development in SOA environments (SDSOA)', ACM, New York, NY, USA, pp. 47–52.

Lomuscio, A. R., Wooldridge, M. and Jennings, N. R. (2003), 'A classification scheme for negotiation in electronic commerce', *International Journal of Group Decision and Negotiation* **12**(1), 31–56.

Michlmayr, A., Rosenberg, F., Platzer, C., Treiber, M. and Dustdar, S. (2007), Towards recovering the broken soa triangle: a software engineering perspective, *in* 'Proceedings of second international workshop on Service Oriented Software Engineering (IW-SOSWE)', ACM, New York, NY, USA, pp. 22–28.

Muthoo, A. (1999), *Bargaining Theory with Application*, Cambridge university press, United Kingdom.

Nassar, T. and Vridhachalam, M. (2010), 'Software as a service: Build a web-delivered saas framework for forms and workflow-driven applications', DeveloperWorks - SOA and Web Services, IBM.
URL: *www.ibm.com/developerworks/library/ar-saasframe/ [Accessed: 1st August 2010]*

Neies, H. and Berman, P. C. (2004), 'Integrating services for old people: A resource book for managers', International Journal of Integrated Care, European Health Management Association.
URL: *www.ncbi.nlm.nih.gov/pmc/articles/PMC1395512/. [Accessed: 1st August 2010]*

Newcomer, E. and Lomow, G. (2004), *Understanding SOA with Web Services*, Phoenix Color Corporation, Maryland, USA.

Nguyen, T. D. and Jennings, N. R. (2005), 'Managing commitments in multiple concurrent negotiations', *Electronic Commerce Research Application* **4**(4), 362–376.

NHS (2010), 'Nhs choices, your health, your choices', National Health Service.
URL: *www.nhs.uk/ [Accessed: 1st August 2010]*

Nitto, E., Penta, M., Gambi, A., Ripa, G. and Villani, M. L. (2007), Negotiation of service level agreements: An architecture and a search-based approach, *in* 'Proceedings of the fifth international conference on Service-Oriented Computing (ICSOC)', Springer-Verlag, Berlin, Heidelberg, pp. 295–306.

Nwana, H. S. (1996), 'Software agents: An overview', *Knowledge Engineering Review* **11**(3), 1–40.

Overeinder, B. J., Verkaik, P. D. and Brazier, F. M. T. (2008), Web service access management for integration with agent systems, *in* 'Proceedings of the 2008 ACM symposium on Applied computing (SAC)', ACM, New York, NY, USA, pp. 1854–1860.

Oxford (2010), 'Oxford advanced learner's dictionary', University of Oxford.
URL: *www.oup.com/elt/catalogue/teachersites/oald7/lookup?cc=global [Accessed 1 August 2010]*

Pahl, C. (2007), 'Semantic model-driven architecting of service-based software systems', *Information Software Technology* **49**(8), 838–850.

Papazoglou, M. P. (2003), Service oriented computing: Concepts, characteristics and directions, *in* 'Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE)', Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, p. 3.

Papazoglou, M. P., Traverso, P., Dustdar, S. and Leymann, F. (2007), 'Service-oriented computing: State of the art and research challenges', *Computer* **40**, 38–45.

Papazoglou, M. P., Traverso, P., Dustdar, S. and Leymann, F. (2008), 'Service-oriented computing: A research roadmap', *International Journal of Cooperative Information Systems* **17**(2), 223–255.

Papazoglou, M. P., Traverso, P., Dustdar, S., Leymann, F. and Krämer, B. J. (2006), Service-oriented computing: A research roadmap, *in* F. Cubera, B. J. Krämer and M. P. Papazoglou, eds, 'Proceedings of Service Oriented Computing (SOC)', number 05462 *in* 'Dagstuhl Seminar Proceedings', Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany.

Paurobally, S., Tamma, V. and Wooldrdige, M. (2007), 'A framework for web service negotiation', *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* **2**(4), 14.

Pistore, M., Traverso, P., Paolucci, M. and Wagner, M. (2009), *From Software Services to a Future Internet of Services*, A European Research Perspective, chapter Towards the Future Internet, pp. 183–192.

Raiffa, H. (1982), *The art and science of negotiation.* Belknap Press of Harvard University Press, Cambridge, Massachusetts.

Raynal, M. and Singhal, M. (1996), 'Logical time: Capturing causality in distributed systems', *Computer* **29**, 49–56.

Rigby, M. (2005), 'Key issues in integrated care plans'. Centre for Health Planning and Management, Keele University, UK.

Rigby, M. (2008), 'Created use case: Discharge planning'. Centre for Health Planning and Management, Keele University, UK.

Rigby, M., Budgen, D., Turner, M., Kotsiopoulos, I. A., Brereton, P., Keane, J., Bennett, K. H., Russell, M., Layzell, P. J. and Zhu, F. (2007), 'A data-gathering broker as a future-orientated approach to supporting epr users', *International Journal of Medical Informatics* **76**(2-3), 137–144.

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).

Rose, S. (1992), *Case Management Social Work Practice*. Longman Publishing Group, London, UK.

Rosenschein, J. and Zlotkin, G. . (1994), *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, MIT Press, Cambridge.

Ruth, P., Jiang, X., Xu, D. and Goasguen, S. (2005), 'Virtual distributed environments in a shared infrastructure', *IEEE Computer* **38**(5), 63–69.

Sierra, C., Faratin, P. and Jennings, N. R. (1997), A service-oriented negotiation model between autonomous agents, *in* 'Proceedings of the eighth European Workshop on Modelling Autonomous Agents in a Multi-Agent World', Springer-Verlag, London, UK, pp. 17–35.

Singh, M. P. and Huhns, M. N. (2005), *Service-Oriented Computing: Semantics, Processes, Agents*, John Wiley and Sons Ltd, Chichester, UK.

Singla, V. (2009), 'The overlapping worlds of saas and soa', Cloud Computing Journal.

URL: *http://cloudcomputing.sys-con.com/?q=node/1047073 [Accessed: 1st August 2010]*

Ströbel, M. and Weinhardt, C. (2003), 'The montreal taxonomy for electronic negotiations', *Journal of Group Decision and Negotiation* **12**(2), 143–164.

Strecker, S. (2004), Multiattribute Auctions in Electronic Procurement - Theory and Experiment, PhD thesis, Universität Karlsruhe (TH), Germany.

Strecker, S. and Seifert, S. (2004), Electronic sourcing with multi-attribute auctions, *in* 'Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS) - Track 7', IEEE Computer Society, IEEE Computer Society, Washington, DC, USA, p. 70165.2.

Su, S. Y. W., Huang, C. and Hammer, J. (2000), A replicable web-based negotiation server for e-commerce, *in* 'Proceedings of the 33rd International Conference on System Sciences', IEEE Computer Society, pp. 4–7.

SunMicrosystems (2009), 'Introduction to cloud computing architecture', Sun Microsystems.
URL: *www.sun.com/featured-articles/CloudComputing.pdf [Accessed: 1st August 2010]*

Tamma, V., Phelps, S., Dickinson, I. and Wooldridge, M. (2005), 'Ontologies for supporting negotiation in e-commerce', *Engineering Applications of Artificial Intelligence* **18**(2), 223–236.

Teich, J. E., Wallenius, H., Wallenius, J. and Koppius, O. R. (2004), 'Emerging multiple issue e-auctions', *European Journal of Operational Research* **159**(1), 1–16.

Teich, J., Wallenius, H. and Wallenius, J. (1999), 'Multiple-issue auction and market algorithms for the world wide web', *Decision Support System* **26**(1), 49–66.

Tierney, B., Becla, J., Gunter, D., Jacobsen, B. and Quarrie, D. (2000), Using netlogger for distributed systems performance analysis of the babar data analysis system, *in* 'Proceedings of Computers in High Energy Physics (CHEP)'.

URL: *http://www-didc.lbl.gov/papers/chep.2K.Netlogger.pdf [Accessed: 1st August 2010]*

Truex, D. P., Baskerville, R. and Klein, H. (1999), 'Growing systems in emergent organizations', *Communications of the ACM* **42**(8), 117–123.

Turner, M., Budgen, D. and Brereton, P. (2003), 'Turning software into a service', *Computer* **36**(10), 38–44.

Turner, M., Zhu, F., Kotsiopoulos, I., Russell, M., Budgen, D., Bennett, K., Brereton, P., Keane, J., Layzell, P. and Rigby, M. (2004), Using web service technologies to create an information broker: An experience report, *in* 'Proceedings of the 26th International Conference on Software Engineering (ICSE)', IEEE Computer Society, Washington, DC, USA, pp. 552–561.

Van Renesse, R., Birman, K. P. and Vogels, W. (2003), 'Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining', *ACM Transaction Computer System* **21**(2), 164–206.

W3C (2010), 'The world wide web consortium', World Wide Web Consortium.
URL: *http://www.w3.org/ [Accessed: 1 August 2010]*

Wallenius, J., Dyer, J. S., Fishburn, P. C., Steuer, R. E., Zionts, S. and Deb, K. (2008), 'Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead', *Management Science* **54**(7), 1336–1349.

Woodall, P., Brereton, P. and Budgen, D. (2007), 'Investigating service-oriented system performance: a systematic study', *Software: Practice & Experience* **37**(2), 177–191.

Yalagandula, P. and Dahlin, M. (2004), 'A scalable distributed information management system', *SIGCOMM Computer Communication Review* **34**(4), 379–390.

Yau, S. S., Ye, N., Sarjoughian, H. S., Huang, D., Roontiva, A., Baydogan, M. and Muqsith, M. A. (2009), 'Toward development of adaptive service-based software systems', *IEEE Transactions on Services Computing* **2**(3), 247–260.

Zhang, L.-J. and Zhou, Q. (2009), Ccoa: Cloud computing open architecture, *in* 'Proceedings of the 2009 IEEE International Conference on Web Services (ICWS)', IEEE Computer Society, Washington, DC, USA, pp. 607–616.

Zhu, F., Turner, M., Kotsiopoulos, I., Bennett, K., Russell, M., Budgen, D., Brereton, P., Keane, J., Layzell, P., Rigby, M. and Xu, J. (2006), 'Dynamic data integration: a service-based broker approach', *International Journal of Business Process Integration and Management* **1**(3), 175–191.

Zuberi, K. M. and Shin, K. G. (1996), A causal message ordering scheme for distributed embedded real-time systems, *in* 'Proceedings of the 15th Symposium on Reliable Distributed Systems (SRDS)', IEEE Computer Society, Washington, DC, USA, p. 210.