Care Planning Through Auction-based Information Negotiation (CAPTAIN): The Ordering of Events in Service Negotiation

Pongpan Pinto
Department of Computer and Information Technology
Faculty of Science, Thaksin University
Phatthalung, Thailand
ppongpan@tsu.ac.th

***Abstract*—The Software as a Service (SaaS) model is a service-based model in which a desired service is assembled, delivered and consume on demand. A 'proof of concept' of SaaS is Information Broker for Heterogeneous Information Sources (IBHIS) which is based on services that deliver data. IBHIS is a fully service-based approach to support the trustworthy integration of heterogeneous forms of information possessed and manipulated by autonomous service providers. Consequently, Care Planning Through Auction-based Information Negotiation (CAPTAIN) is developed to extend the concepts and role of the broker as used in IBHIS. In particular, CAPTAIN also extends the concepts and role of the service negotiation function to demonstrate a full range of service characteristics: description, discovery, composition, negotiation and delivery.**

A service-oriented broker architecture for CAPTAIN has been developed based on a case study within healthcare context. It includes a negotiation model developed for the service negotiation to deals with the needs of a client and the constraints of service providers within planning service oriented circumstances. A 'proof of concept' of CAPTAIN is demonstrated as a service-oriented planning broker system that is flexible, adaptable and platform independent. The CAPTAIN system is a distributed system consisting of several different system components which are spatially separated. Hence, this paper will present the investigation of the system behaviour by observing a partial ordering of the events in the CAPTAIN system. The events occur when there are the execution of a subprogram or interactions among the system components. The correct ordering of the events can ensure that the system components are accurately regulated by the negotiation process and protocol. The timestamp of a sequence of events is recorded by using an event logger. The results of our study show that all system components of CAPTAIN can perform their tasks and interact with each other correctly during service negotiation between the planning broker and the service providers. The system components exchange their messages by sending and receiving among them, following the process of service negotiation serial. As a result, the CAPTAIN system can provide planning services properly to produce the integrated care plan for the client.

Keywords-component; Software as a Service; IBHIS; CAPTAIN; Web Services

I. INTRODUCTION

The conventional development of software has focused on supply-side issues, driven by developers or technology rather than end users. [3] In addition, software developed and delivered as a product is not adaptable to the new era of rapidly changing business needs. Moreover, the internet has been progressively used as a disruptive platform for new various types of business model. [8] As a result, the software industry as established by software vendors has begun to move from software products to more profitable software services for the end users. [7] To deal with new demands of the end users, software can be delivered as a service, focusing on the rapid changing needs of the end users or service consumers. [12][4]. Therefore, the "service-oriented" approach leads to the benefit of the independent from particular software programming languages or operating systems. [9] This 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. [3][1][6]

The Software as a Service (SaaS) is a service-based model. It is based on a demand-led concept in which a desired service is assembled, delivered and consumed on demand. By using the SaaS model, end-user services are composed out of smaller ones (and so on recursively), procured and paid for on demand. Users can generate, compose and assemble a service by gathering services from a number of service suppliers in order to achieve their needs at a specific point in time. The SaaS model consists of three layers: service description, service integration and service transport. The service integration layer comprises five main service functions: description, discovery, negotiation, delivery and composition.

The Care Planning Through Auction-based Information Negotiation (CAPTAIN) system is a service-oriented software providing healthcare planning broker services for the user. The architecture of CAPTAIN has been derived from Information Broker for Heterogeneous Sources (IBHIS) and developed based on the SaaS model. The CAPTAIN system is a distributed system consisting of many diverse system components which are spatially separated. The distributed CAPTAIN system is needed to be observed the ordering of events. Therefore, this paper will present the investigation of a partial ordering of the events occurring when there are the executions of a subprogram or interactions among the system components. The correct ordering of the events can ensure that all system components are accurately regulated by the negotiation process and protocol. As a result, the CAPTAIN system can perform correctly to produce the result for a client.

The remainder of this paper is structured as follows. Section 2 and 3 introduce the background of the IBHIS and CAPTAIN respectively. Section 4 describes the theory related to the ordering of events in distributed systems that is used for the investigation of the CAPTAIN system's behavior. Section 5 and 6 presents the research method and, then, the result and discussion of this paper respectively. Finally, section 7 presents some conclusions and orientations for future work.

II. INFORMATION BROKER FOR HETEROGENEOUS SOURCES (IBHIS)

The IBHIS broker is the demonstration as 'A proof of concept' of the overall set of SaaS concepts in a form that was based on services that deliver data. [3][4] IBHIS explored a fully service-based approach to large-scale healthcare data integration, gathering information from distributed, heterogeneous and autonomous data sources.

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.

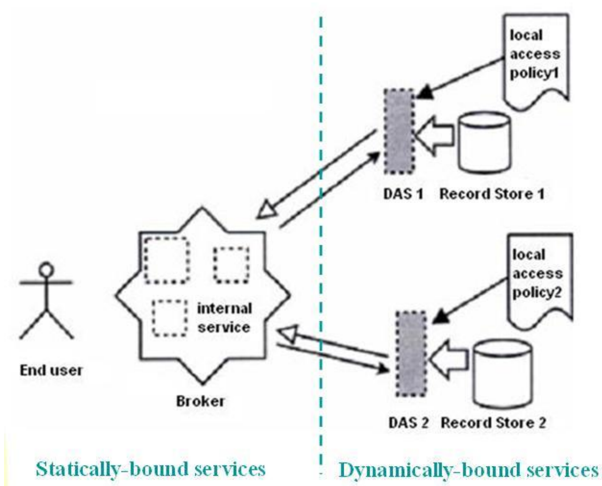

Figure 1 The System Architecture of IBHIS

Figure 1 presents the IBHIS architecture that consists of basic services used in two forms: statically-bound service and dynamically-bound service.

- *The statically-bound services* are the set of services that are employed within the information broker to provide functional tasks, such as user interface, user authorization services and query formulation.
- *The dynamically-bound services* are the set of services, termed as Data as a Service (DaaS). The DaaS is a service supplying information, used where the set of related information sources is usually regulated dynamically. According to a query specified by a client or an end user, the information broker can dynamically locates and accesses distributed data sources, provides as data services, selecting these according to the information that they can provide.

  The dynamically-bound services, needed to perform a task, are determined, located and bound at the time of execution. The concept of a Data Access Service (DAS) is used to form an interface between the 'physical' structure of the data sources, and the broker itself.

The IBHIS system performs by supplying information services to meet the needs of the end users. It delivers integrated data that are gathered from distributed, heterogeneous and autonomous data sources. 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 or providers. Within the structure of UK health and social care, this could provide an organisation changing context which could then be used by various end users for diverse purposes.

III. CARE PLANNING THROUGH AUCTION-BASED INFORMATION NEGOTIATION (CAPTAIN)

A. The Architecture of CAPTAIN

The architecture of CAPTAIN has been derived from IBHIS. Figure 2 presents its architecture that is composed from four major service components: the planning broker (P-Broker), the information broker (I-Broker), data access services (DASs) and semantic registry. The P-Broker and the I-Broker are statically-bound services while the DASs are dynamically bound services.

- *The DASs* are a set of heterogeneous, autonomous and distributed data sources owned by several healthcare service providers. The DASs supply healthcare information related to the integrated care plan, for instance service description and healthcare professionals, to the I-Broker.

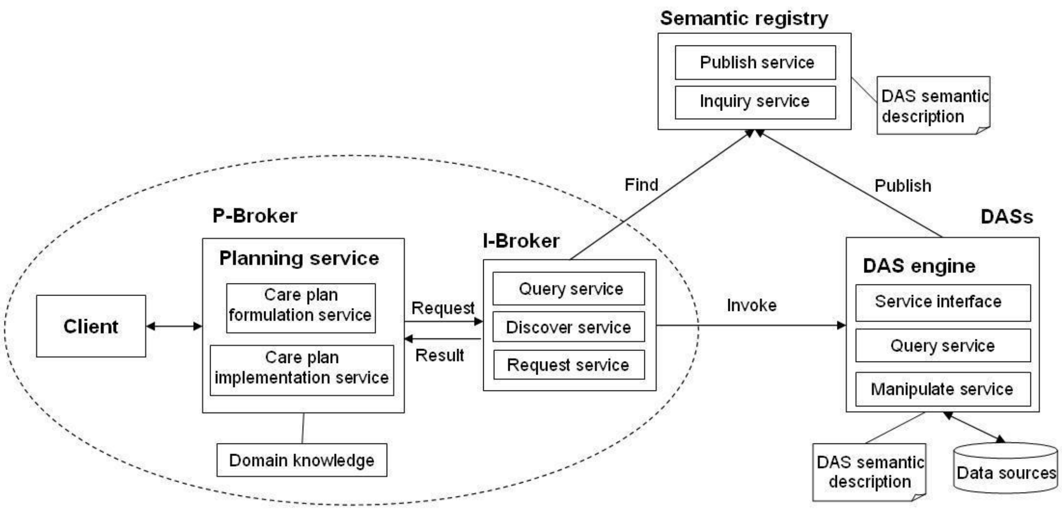

Figure 2 The System Architecture of CAPTAIN

- *The I-Broker* performs as an 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. According to the request or interests of the P-Broker, the I-Broker takes action on behalf of the P-Broker to gather information from the DASs. Then, the I-Broker blends the information from the DASs into the composite form defined by the P-Broker. Finally, the I-Broker returns the result to the P-Broker.
- *The P-Broker* provides a care planning service by producing the integrated care plan according to the needs of the end user or client. The P-Broker uses data and information supplied by the DASs via the I-Broker, together with by its planning knowledge repository called "*domain knowledge*". The P-Broker employs a "*Planning service*" to generate the integrated care plan for the end user. The "Planning service" consists of two main services: "Care plan formulation service" and "Care plan implementation service".
  - *The "Care plan formulation service"* is used to analyse the end user's request and, then, formulate the requests or queries for the I-Broker.
  - *The "Care plan implementation service"* uses the requests from the "Care plan formulation service" to gather and analyse the information from the DASs, via the I-Broker, in order to create the integrated care plan for the end user.
- *The semantic registry* contains information about the DASs' service descriptions. It consists of two key service components. The "*Publish service*" and "*Inquiry service*" are used by the DASs and the I-Broker to publish and find the DASs' service description respectively.

B. Negotiation Model

CAPTAIN includes the negotiation model to deal with the negotiation situation between the negotiation participants to agree on the terms and conditions relating to the supply of the services. As shown in figure 3, the negotiation model for CAPTAIN consists of four main elements: negotiation process, negotiation object, negotiation protocol and decision model. During service negotiation, the negotiation elements are used by the system components of CAPTAIN, especially the P-Broker, the I-Broker and the DASs, in order to create the integrated care plan the meet the needs of the end-user.

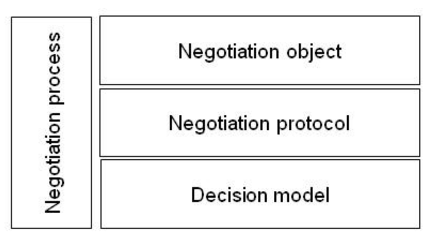

Figure 3: The negotiation model of CAPTAIN

- *The negotiation process* is a dynamic end-to-end process. It consists of three main sequential phrases of negotiation process. It begins from the end-user request and ends up with a resolution.
  - *The pre-negotiation phrase* is concerned with producing the initial information, such as DASs' service description and the query of the end user, for the service negotiation phrase.
  - *The service negotiation* phrase is the phrase in which the P-Broker negotiates with the DASs

via the I-Broker in order to produce the integrated care plan for the end user.

  - *The post-negotiation* phrase deals with the negotiation results from the service negotiation phrase. The end user is involved in the decision making since the end user accepts or declines the results.

- *The negotiation object* contains the information required for negotiation between the P-Broker and the DASs. There are three main types of negotiation object: query, offer and acknowledgement. The P-Broker creates the query and proposes it to the DASs. Then the DASs respond to the P-Broker's query with the counter offers. Finally, the P-broker issues the acknowledgement of its decision-making according to the counter offers to the DASs.

- *The negotiation protocol* defines the rules and the states of negotiation interaction followed by all system components of CAPTAIN during service negotiation process. So the P-Broker and the DASs can negotiation to reach a mutual agreement between them. The negotiation protocol is shown in figure 4.

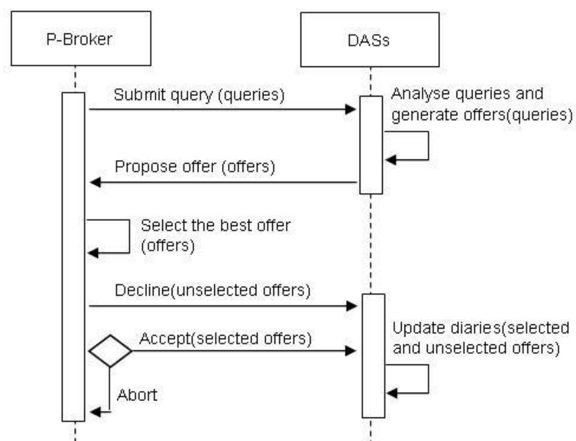

Figure 4 The Negotiation Protocol

- *The decision model* defines the rules employed by the P-Broker and the DASs for decision making. Each negotiation participant has its own decision model of strategy which is used to define the plan of decisions or actions needed to achieve its negotiation objective.

C. Service Negotiation in CAPTAIN

Figure 5 presents a sequence message diagram that illustrates service negotiation in CAPTAIN. It includes all negotiation elements for CAPTAIN to support the service negotiation between the P-Broker and the DASs. The result of the negotiation is the integrated care plan that meets the needs of the end user.

- The pre-negotiation phrase produces the information needed for the service negotiation phrase. The DASs publish their service descriptions to the semantic registry so that the I-Broker, then, can find their service descriptions during service negotiation between the negotiation participants. After that, the end user generates a request for the P-Broker by choosing and revising a care plan retrieved from the domain knowledge. Finally, the P-Broker proposes the request to the I-Broker to formulate specific queries for the DASs.

- The service negotiation phrase takes place between the P-Broker and the DASs. The P-Broker proposes the queries, via the I-Broker, to the DASs for the DASs' counter offers. The P-Broker selects the counter offers preferred by the end user. Then the P-Broker creates and presents the integrated care plan to the end user.

- The post-negotiation phrase involves the decision-making by the end user to accept or decline the result from the P-Broker. The accepted integrated care plan is updated to the P-Broker's domain knowledge. Then the P-Broker will issue an acknowledgement of the end user's decision making to the DASs to update data in their data sources.

D. Implementation

The CAPTAIN system is a dynamic web application. The development and implementation of the CAPTAIN broker prototype is based on Java 2 Enterprise Edition (J2EE) Web Services technologies, and runs within the Eclipse Java EE IDE for Web Developers environment that works on a Windows platform. The system prototype supports Web Services standards, for example XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description, Discovery and Integration).

The CAPTAIN system 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.

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 integrated care plan is used to provide a research case study. A clinician, acting on behalf of the patient, use the CAPTAIN system to negotiate with different healthcare service providers in order to satisfy the form of the patient's needs. The aim of the negotiation is to agree on the terms of any conditions relating to supplying the healthcare services for the patient.

IV. Ordering of Events in Distributed Systems

In a distributed system, the ordering of events or processes is one of the basic aspects for an observer to analyse and understand the system for such purposes such as performance analysis, monitoring and debugging the system.

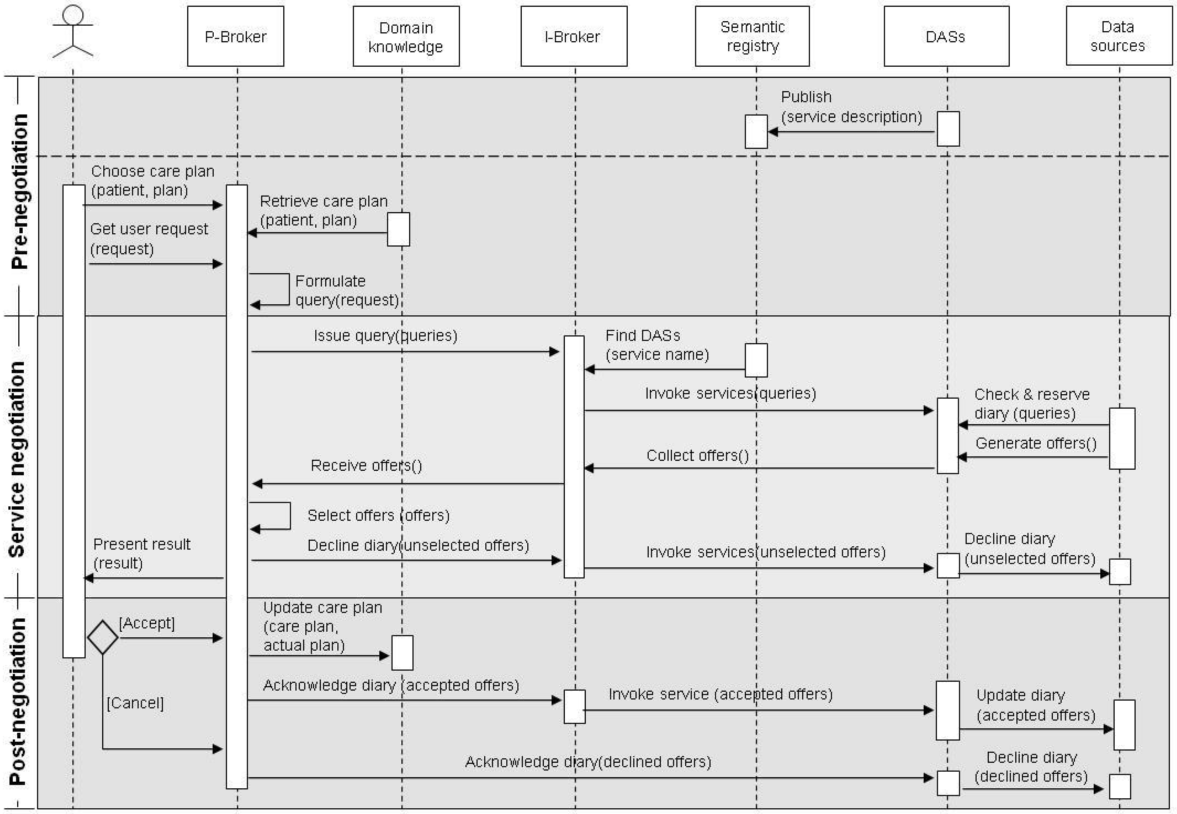

Figure 5 The Sequence Diagram of Service Negotiation in CAPTAIN

[2][11] A distributed system consists of a set of diverse processes that are spatially separated. Each process comprises a serial of events. Based on an application, the events in the process are mostly categorized 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

A partial ordering of the events is one of the common approaches that is used to observe the ordering of events in a distributed system. [9] For any single process, the order of the events with a priori overall ordering describes how an event takes place before or after the other event. In a specific order, 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 → 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 → b.

- If a → b and b → c, then a → c.

Hence, the events or discrete actions is useful for understanding or analysing any multi-process system by providing the means for the observation of what is occurring in the system. [2]

V. Research Method

The aim of this research is to investigate the system behavior of the CAPTAIN system during service negotiation between a planning broker and service providers. For the observation of the system behavior of CAPTAIN, *the event logger* is used to record a partial ordering of the events which are followed by negotiation process and protocol of the CAPTAIN system. It 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'.

Figure 6 illustrates the system diagram of the event logger. 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 kinds of event data:

- the names of the system component
- the event together with its timestamp.

The example of the event data is shown as follows:

event('P-Broker', 'offer', 'I-Broker', '2012-03-15 09:47:13.908')

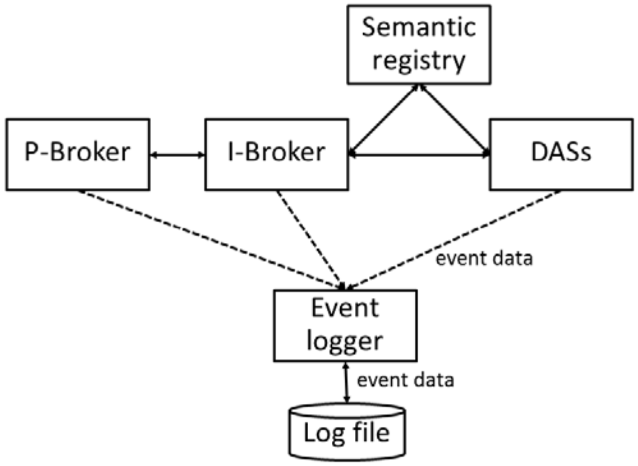

Figure 6 System Diagram of Event Logger

This event occurs 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 produce timestamps to millisecond precision.

VI. RESULT AND DISCUSSION

Table 1 shows the event data of the event logger recording a partial ordering of the events which are followed by negotiation process and protocol of the CAPTAIN system. In addition, figure 7 presents the sequence diagram of the sequence and timestamps of the events of the system components interacting each other during the service negotiation process.

Sender	Message	Receiver	Timestamp
P-Broker	Query	I-Broker	2012-03-15 09:47:15.628
Semantic registry	DAS list	I-Broker	2012-03-15 09:47:16.642
I-Broker	Offer	DAS1	2012-03-15 09:47:17.018
DAS1	Counter offer	I-Broker	2012-03-15 09:47:18.078
I-Broker	Offer	DAS2	2012-03-15 09:47:18.520
DAS2	Counter offer	I-Broker	2012-03-15 09:47:35.19.465
I-Broker	Offer	DAS3	2012-03-15 09:47:20.189
DAS3	Counter offer	I-Broker	2012-03-15 09:47:21.060
I-Broker	Result	P-Broker	2012-03-15 09:47:21.643

Table 1 The Event Data of the Event Logger

The ordering of events begins 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 specific time and a particular point of the each event. Next, the I-Broker exchanges the messages by sending and receiving to and from several DASs of data service providers according to the ordering of negotiation process and protocol. Finally, the P-Broker receives the result from the I-Broker to produce the integrated care plan for the end-user. These events form the serial events in timestamp order. As a result, all system components process their tasks, and then, send and receive messages among them as in orders specified by CAPTAIN.

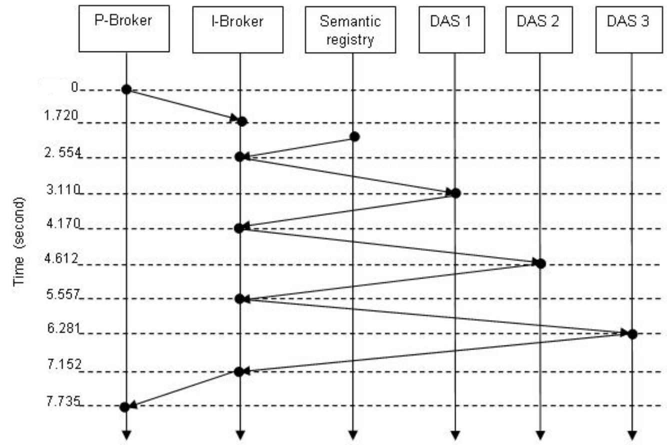

Figure 7 The Ordering of Events in Service Negotiation for CAPTAIN

From the aspect of service negotiation in CAPTAIN, the order of the events with a priori total ordering illustrates that an event takes place before or after the other event. Hence, the proper ordering of the events entails that the main system components of the CAPTAIN system are properly regulated by the negotiation process and protocol. The negotiation objects or messages are exchanged by sending and receiving between the P-Broker and the DASs via the I-Broker as in the serial of processes of service negotiation. As the result, the P-Broker can receive the negotiation result to produce the integrated care plan that meets the needs of the end user.

VII. CONCLUSION

In this paper, The IBHIS and CAPTAIN systems are described that both of the systems have been developed based on the concept of service-based software model. The CAPTAIN system is a distributed system consisting of a number of diverse system components which are spatially separated. The partial ordering of events in the CAPTAIN system was observed by using an event logger. The result of the investigation shows that all system components are exactly regulated by the negotiation process and protocol. They can exchange the negotiation objects sent and received among the system components as in the order of processes of the negotiation process and protocol. Therefore the CAPTAIN system performs properly to provide planning services by producing the integrated care plan for the client.

As a future work, the event logger should be developed to investigate the system behavior of CAPTAIN, not only for the service negotiation phrase but also for the pre-negotiation and post-negotiation phrases. Therefore we can ensure that the CAPTAIN system can perform correctly from the end user request to a resolution.

ACKNOWLEDGMENT

I would like to acknowledge and extend my gratitude to Professor David Budgen for his guidance and constructive advice as well as his encouragement and support when I did my PhD research at University of Durham, UK. I would like to thank Professor Michael Rigby who supported me fruitful knowledge for my research and Dr. Mark Turner who had supported me, particularly in the part of system implementation. Both of them work in Keele University, UK.

REFERENCES

[1] A. Elfatatry, P. and Layzell, "Negotiating in service-oriented environments", Commun. ACM, 2004, 47(8), pp. 103–108.

[2] C. Fidge, "Fundamentals of distributed system observation", IEEE Software, 1996, 13(6), pp. 77–83.

[3] D. Budgen, P. Brereton, and M. Turner, "Codifying a service architectural style", in COMPSAC, 2004, pp. 16–22.

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

[5] D. Budgen, M. Turner, I. Kotsiopoulos, F. Zhu, M. Russell, M. Rigby, K. Bennett, P. Brereton, J. Keane, "Managing healthcare information: The role of the broker", Healthgrid, 2005, pp. 3–16.

[6] H. Demirkan, R. J. Kauffman, J. A. Vayghan, H. G. Fill, D. Karagiannis, and P. P. Maglio, "Service-oriented technology and management: Perspectives on research and practice for the coming decade", Electronic Commerce Research and Application, 2008, 7(4), pp. 356–376.

[7] IBM-SOA, "IBM:Service-oriented architecture (SOA)", [Online] Available from: http://www.01.ibm.com/software/solutions/soa/ [Accessed: 1st August 2012].

[8] J. Domingue, D. Fensel, J. Davies, R. González-Cabero, and C. Pedrinaci, "The Service Web: a Web of Billions of Services", A European Research Perspective, 2009, pp. 203–216.

[9] L. Lamport, "Time, clocks, and the ordering of events in a distributed sys"em", Communication, ACM, 1978, 21(7), pp. 558–565.

[10] M. Papazoglou, P . Traverso, S. Dustdar, and F . Leymann, "Service-oriented computing: State of the art and research challenges", Computer, 2007, 40, pp. 38-45.

[11] M. Raynal, and M. Singhal, "Logical time: Capturing causality in distributed systems", Computer, 1996, 29, pp. 49–56.

[12] T. Kohlborn, A. Korthaus, and M. Rosemann, "Business and software service lifecycle management", Enterprise Distributed Object Computing Conference, IEEE International, 2009, pp. 87–96.