Journal of Software Engineering Studies
Vol. 2, No. 4, 161-172, June 2008

the Third Taiwan Conference on Software Engineering (TCSE07)

A Study on Intention-Aware Service-Oriented Systems

Chiung-Hon Leon Lee
Department of Computer Science and Information Engineering,
Chung Chou Institute of Technology, Taiwan
Alan Liu
Department of Electrical Engineering and Center for Telecommunication Research,
National Chung Cheng University, Taiwan

A novel approach is proposed to cope with the problems of how to interpret the service request intention from the user's entered string and how to integrate the internal services implemented in the system and external services distributed on the Internet to satisfy the user's request. We identify these two problems as the user intention extraction and the user intention satisfaction, respectively. If a service-oriented system with the abilities of user's intention extraction and satisfaction, we call it an intention-aware service-oriented system (IASOS). We start the system design from the view of system requirements engineering. The requirements specification is generated by the goal-based requirements analysis in which the functional and nonfunctional requirements will be extended with goal models. The designer designs the system capabilities based on the requirements specification. Based on the proposed intention extraction approach, the user's vague and imprecise intention will be extracted and mapped to computer understandable and computable goal models. The hierarchical task network (HTN) planning approach is used to generate a plan for satisfying the extracted user intention. An architecture is designed for demonstrating how to integrate the intention extraction and satisfaction methods into applications.

Keywords: intention-aware system, service-oriented system, goal models, planning

1. Introduction

A long-term goal of computer systems is to provide users with more variant and richer services in which requesting services from the computer system is as easy as call for help from the human assistant. Recently, service-oriented computing (SOC) [1, 2] is emerging as a new and promising computer paradigm that centre on the notion of service as the fundamental component for developing distributed software applications. The loosely coupled and reusable characters of the Web service make it as a good choice to enhance the service capabilities of the computer system.

How to deliver proper services to satisfy the user's request is an important research issue of service-oriented systems, especially when the system expected to serve the user automatically with less user interaction. In this paper, we discuss the problems of how to interpret the service request intention from the user's entered string and how to integrate the internal services implemented in the system and external services distributed on the Internet to satisfy the user's request. These two problems are identified as the user intention extraction and the user intention satisfaction, respectively.

If a service-oriented system with the abilities of user's intention extraction and satisfaction, we call it an intention-aware service-oriented system (IASOS). An IASOS provides a more convenient and efficient way for the user to use the services of a computer system. The operation concepts of an IASOS are shown in Figure 1.

In our approach, an intention I is represented by a vector space which consists of a set of terms extracted from the user's entered string. In other words, $I = \{T_1, T_2, ..., T_n\}$, in which T indicates terms in the service request string and n is the number of terms. A goal G is an abstract description of the capability which the system can performed. The intention extraction *IE* problem can be described as *IE* : $I \rightarrow G$.It means that given an intention I, after the *IE* process, we can map I to an abstract description of system capability G.

A plan P is composed by a series of system operations. A system operation *OP* describes a single or a composite system function. The operations can be used to invoke internal or external services. An internal service is a system function which can be used to serve the user or can be combined with external services to generate more complex services. An external service is the service provided by external objects which could be a Web service provider or other software agents [3]. The intention satisfaction *IS* problem can be represented as *IS* : $G \rightarrow P$. It means that given a goal G, after the IS process, the system can derive a plan P for achieving G. How to execute generated plan and monitoring the execution of the plan are also important problems when constructing an intention-aware system. However, these two problems are out of the scope of this paper.

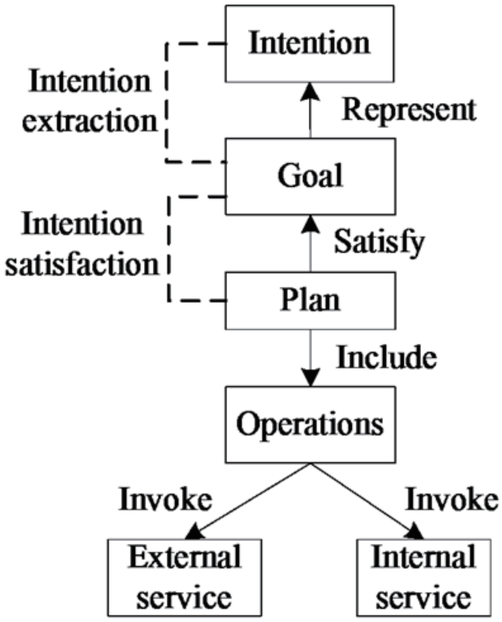


Figure 1. The concept of intention extraction and satisfaction.

There are two main contributions of this paper. First, we model user interactions with service-oriented systems as intention-driven activities. Second, we proposed an approach to integrate internal system functions, external Web services, and user information to satisfy the user's service requests. The proposed approach allows a Web service system to pool knowledge about application domain, system capabilities, Web services, and user information to satisfy extracted user intentions. In other words, we draw attention to the parallel between intention-based human-computer interaction (HCI) [4] systems and the process of selecting, composing, and executing Web services.

2. BACKGROUNDS

Web services

Web services are a good choice for loosely coupled architectures and reusable software components which can be published, located, and invoked via the Internet. There are several technologies such as SOAP, WSDL, and UDDI are developed to enable Web services. However, to fully satisfy the requirements of business applications, the current technologies still have shortcomings such as security, composition, and semantic problems [5].

One of the most attractive characters of the Web services is service composition that the atomic services can be future assembled into value-add composite services to solve more complex problems. There are several proposals for creating services compositions in a standardized and systematic fashions such as Web Services Flow Language [6] and BPEL4WS [7]. However, these standards describe Web services content in terms of XML syntax which lacks both a well defined semantics and sufficient expressive power.

The Semantic Web services [8] solve problems on the semantic level of Web services and address Web services descriptions as a whole. The semantic markup language such as OWL-S and its previous release DAML-S [9] are proposed to describe the capabilities and content of the Web services in a computer interpretable language and improve the quality of service discovery, invocation, composition, monitoring, and recovery.

Intelligent agent

An agent is a software entity that has some properties of a human such as autonomy, reasoning, learning, and knowledge level communication [10]. To facilitate the service access, the agent technology is widely used in the Web service research [8, 11, 12, 13]. The agent represents the user to discover, interact, and compose Web services to satisfy the user's goals or intentions.

Since the agent has to delegate the user to do something and to serve the user, how to make the agent understanding the user's goal or purpose becomes an important function of the system. In [14, 15], we proposed an approach to model the user's intention with goal that makes the service requester intention-aware. We continue our previous to explore research issues of constructing an intention-aware system.

Challenges of Web services provision

Although there are several techniques and standards have been proposed to facilitate the Web services access, to a Web services system, wishing that the results returned from the service providers can be used directly to satisfy the user's request is still an open problem. The system might have to interact with the user to elicit more information for acquiring the suitable Web services, discover services, select services, compose services, process information returned from the services, and demonstrate the results to the user. For these purposes, how to integrate the Web service, system's capabilities, and elicited user information becomes an important issue when developing an intelligent Web services application. An interaction scenario among the system user, service requestor, service broker, and service provider is illustrated in Figure 2.

The service requester interacts with the user and tries to complete the user's delegations. The service broker collects a list of Web services advertisements from the service provider, accepts the service request from the requester, and tries to discover, select, and access the Web service to satisfy the service request from the requester.

Because the requester has no knowledge about the service providers, the initial query it sends to the broker might not correspond exactly with the query input that the provider might require in order to provide the service. Sometimes, the requester might have to interact with the user to elicit more information for accessing the Web services. The discussion of the broker protocol can be found in [13]. In this paper, we focus our discussion on the service requester.

For example, assume that a user wants to buy a book via bookstore services; the system has to provide a mechanism to understand that the user wants to buy a book. Next, if the user has no detail information about the book, the system has to discover a service to help the user acquiring detail information of the book such as ISBN, correct book title, publisher, or provider etc. Then, the system has to display the queried book information to the user, elicits which

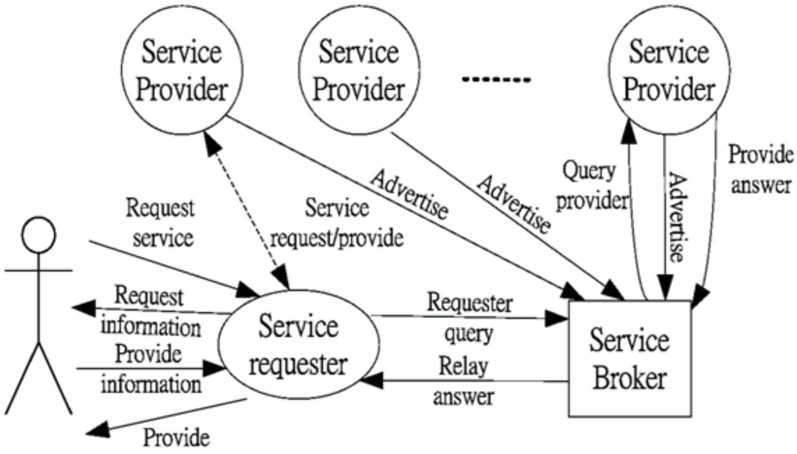


Figure 2. An interaction scenario of the user, requester, broker, and provider.

book is what the user really wants, and discovers a service for book ordering. If there is more than one choice of the same book, the system can also sort the found book by their price for getting inexpensive one. Finally, a book order will be returned and displayed to the user.

In this simple bookstore example, we understand that the user's request cannot be satisfied directly if only compose query book and order book services which provided by the service provider. The system has to interact with the user for getting more book information and future processes the book list returned from the service provider. This motivates us to propose an approach to integrate the Web services, system function, and user's information.

3.SYSTEM DESIGN requirements

Because the Web services requester has no priori knowledge about the service providers and Web services, how to map the service request to related Web services is an important challenge in Web services research. In software requirements engineering [16], the requirements is elicited from the system user and the system will be designed to provide a set of services to achieve the elicited requirements.

We start the design of an IASOS from the view of system requirements engineering. In system construction phase, the requirements specification is generated by goal based requirements analysis in which the functional requirements will be extended with goal models [17, 18]. We called the user's request in using a system as a user-specific goal, and the system-specific goal is the requirement on services that the system provides. The original goal derived from a service request is always a user-specific goal because it reflects the request of the user.

For example, consider three goals as follows:

"buy an inexpensive java book from bookstore,"

"query a java book," and

"sort booklist by price."

The first is a user-specific goal. The goal is initiated by the user and has to be satisfied or fail. The next could be a user-specific or a system-specific goal, because the query book goal could be used to serve the user or to support the first goal. The third is a system-specific goal, because the goal is initiated by the system to support the first goal to find an inexpensive book about java programming.

The designer designs the system capabilities based on the requirements specification. A set of user-specific goal models that represents the user requirements is selected and refined as the basis for mapping the user requests to the system functions. The system specific requirements are also represented by a set of system-specific goal models to extend the user-specific goal models. A goal model might include or extend by other sub-goal models. The relationships between original goal models and sub-goal models can be derived from software requirements analysis. The concept of modeling service delivery as intention extraction and satisfaction activities is shown in Figure 3.

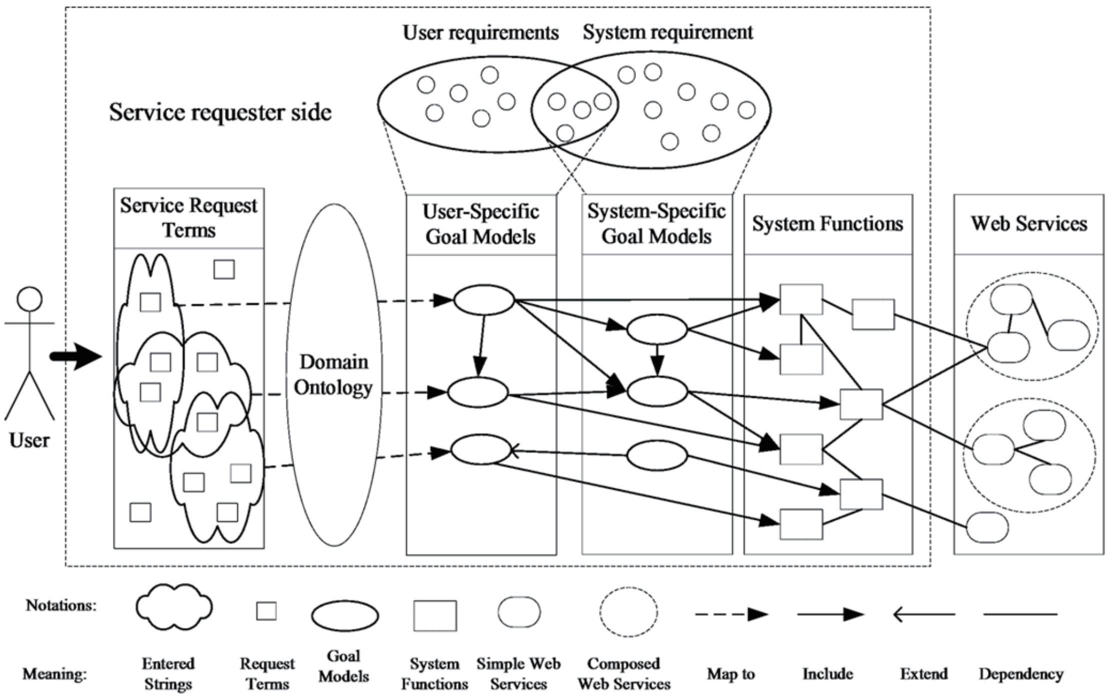


Figure 3. Mapping service requests to system functions.

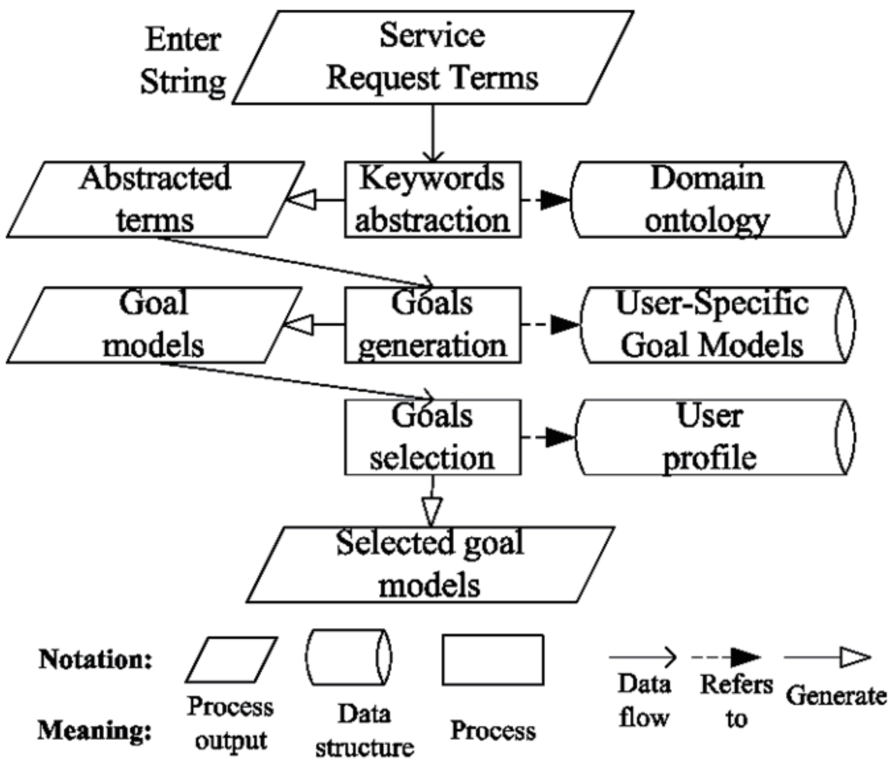


Figure 4. User request extraction processes.

In our approach, users request services by entering verb-based strings. For example, an entered string could be "Buy a java book from Amazon." Each term in the entered string is a service request term. The verb "buy" indicates the action of what the user wants the system to perform and the other terms will be classified as parameters of the action. A service request from the user is composed by a set of service request terms. We assume that the service request implies the user's intention. By using predefined domain ontology and information of goal models, the services requests will be mapped to a user-specific goal model in the goal structure. The goal structure consists of goal models and their relationships; it is used to bridges the gap between service requests and system capabilities.

Figure 4 shows the procedure of how to map user's entered strings to user-specific goal models. Three processes, the keywords abstraction, the goal generation, and the goal selection, are designed to perform the mapping processes. The domain ontology, goal structure, and user profiles are used to support the processes. The keywords abstraction process receives the service request terms from the user and refers a lexical dictionary and domain ontology to generate a word sense set. The goal generation process uses the annotated, abstracted terms and the predefined user-specific goal models for matching the candidate goal models. If there is any verb which exists in the terms, the system will use the verb and synonyms of the verb as index to search the goal structure to find the matching goal models. The goals selection process receives candidate goal models and selects a reasonable goal model by referring user preference record.

When an original goal model is selected to represent the user's intention, the related sub-goal models of the original goal model is also retrieved to represent the implicit intention of the user. Base on the goal models, the system can generate a plan composed of a set of actions to satisfy the user requests. When the plan is executing, the system might interact with service broker or provider to call for external services to complete the plan or interact with the user to get more information for requesting services. The whole process of an intention-aware system is like a requirements reuse approach. The system user enters a phrase to indicate user's requirement, and the system will retrieve a previous elicited user-specific requirement and related system capabilities for execution. If to complete the user's request exceeds the system capabilities, the system can calls external service broker or provider to extend its capabilities.

4. IMPLEMENTATION

Intention Interpretation by Case-Base Reasoning

Providing users with an environment in which interaction with computers is as easy as interaction with human beings is a long-term goal of natural language processing (NLP) [19, 20]. There are many research issues in NLP and many techniques proposed to handle these issues such as using augmented finite state machines to parse sentence, translating parsed results into SQL, predicate calculus, or equivalent semantic networks, applying spelling and grammar checker to facilitate paper editing, and using keyword-based information retrieval techniques to retrieve information from large amounts of data. However, how to use natural language to query or command computers is still one of big challenges in computer science researches because the user might overshoot or undershoot the capabilities of natural language interface [21].

In IASOS, we use case-based reasoning (CBR) approach for the user intention interpretation [22, 23]. The extracted service request will be mapped to a set of goal models [14] that are elicited in a goal discovery process. The advantage of using CBR approach is that the previous request interpretation experience can be stored in a Case-Base (CB), when the system encounters a new service request, the old interpretation experience of service request can be retrieved and adapted to interpret the new service request. Since the new problem has been solved successfully and CB has no similar case, the system can store the new solution to CB for future usage.

We store goal models and their related service request information into CB and develop a novel similarity measure approach for mapping the user's service request to the case. An adaptation-knowledge base is developed to adapt the retrieved case. Finally, goal models could be extracted form the adapted case to represent the user's service request.

The relationships among the case-base, case, problem domain, and solution domain are shown in Figure 5. A case of our approach consists of three parts: the description part, solution part, and relation part. The description of the problem will be put in the description of the case here and later we will use this part for similarity measures during case retrieval. The descriptions usually are the feature set of the problem. Various representation methods can be applied here like lists, semantic networks, frames and objects. Sometimes, scenarios are also used here. We use the original service request extracted from requirement specification or entered by the user as the description part.

Intention Satisfaction by AI-Planning

Since the unclear service request string has been interpreted and a related goal model is generated to describe the request, the next problem is how to satisfy the goal. We use automated planning techniques developed in Artificial Intelligence (AI) researches [24] to deal with this problem.

Although the services composition problems have been discussed in many literatures, most of proposed approaches are based on the service broker's and provider's point of view not in the service requester's point of view. In many cases, the user's request is assumed that has been acquired and represented correctly. However, how to acquire the user's request is important if the system wants to provide more convenient or personalized services. The requester agent might have to interact with the user to elicit more information for acquiring the suitable Web services, perform skills for processing information returned from the services broker or provider, and demonstrate the results to the user.

In our approach, we consider the services composition problem from the service requester's point of view and focus our discussion on how to integrate the Web services returned results, agent's skills, and elicited user information for user intention satisfaction.

A plan consists of a series of actions that should be performed for achieving the related goal. By using the information of the goal, the execution state of the system, and a well designed plan library, the system can retrieve and adapt a series of actions from the plan library for achieving the user's goal. If

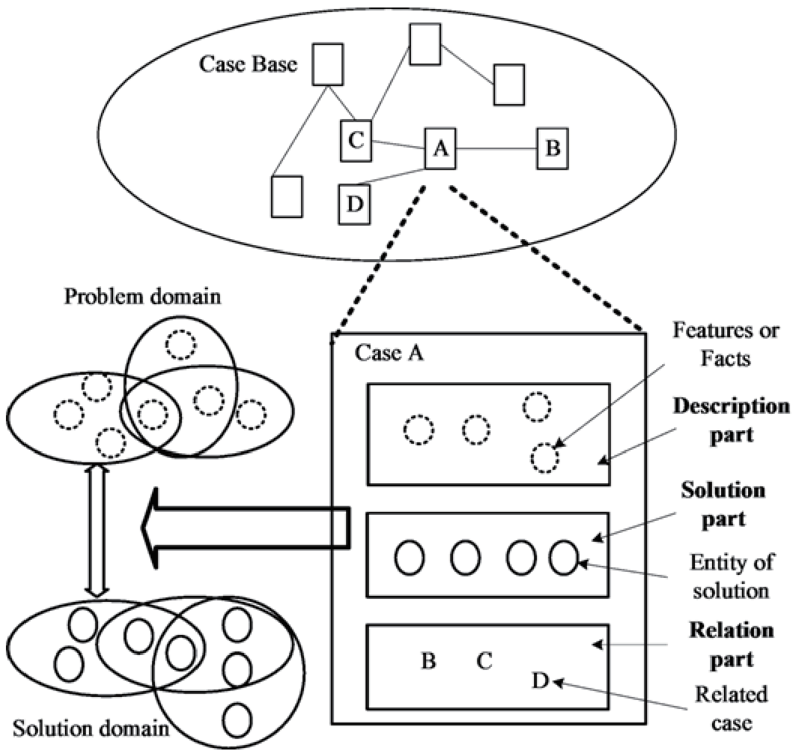


Figure 5. The relationships among the case-base, case, problem domain, and solution domain.

some successful or failed results are produced by the execution of a plan, these messages could be passed to the system. If a soft original goal of the user fails, the partial result completed by the sub-plans can also be displayed to the user.

A goal-plan structure is shown in Figure 6. The structure is divided into abstract level and concrete level. The abstract level contains the relationship and descriptions of plans and actions. The concrete level includes different concrete functions performed by the system.

A plan is composed of several actions. An action might generate sub-plan or call a concrete system function to complete the requirements of the action. We define that the action to generate sub-plan as an abstract action, and the action to call the predefined function in the concrete level as a concrete action. If a plan contains one or more abstract actions on the leaf of the plan hierarchy node, it is an abstract plan; otherwise it is called a concrete plan. The descriptions of the action decomposition methods are also stored in the plan library.

The concrete level contains three types of functions performed by the system: Web service, human skill, and agent skill. The function of the Web service is to acquire results from the service provider or broker, the function of human skill means to elicit required information from the user for services invocation, and the function of agent skill is to trigger the agent to perform some actions to process the data acquired from the Web services or the user.

The original goal could be any user-specific goal which predefined in the goal model and the action generation depends on the situation when the system is executing. The advantages of such hierarchy are the reusability and flexibility. It should be noted that in the plan, we are only interested in the integration of Web service, human information, and the agent skill. The problem of service composition is assumed to be performed by the service broker agent because we assume that the requester agent has no information about the services and has to cooperate with the broker or provider agents for satisfy the user requests.

There are various planning methods which have

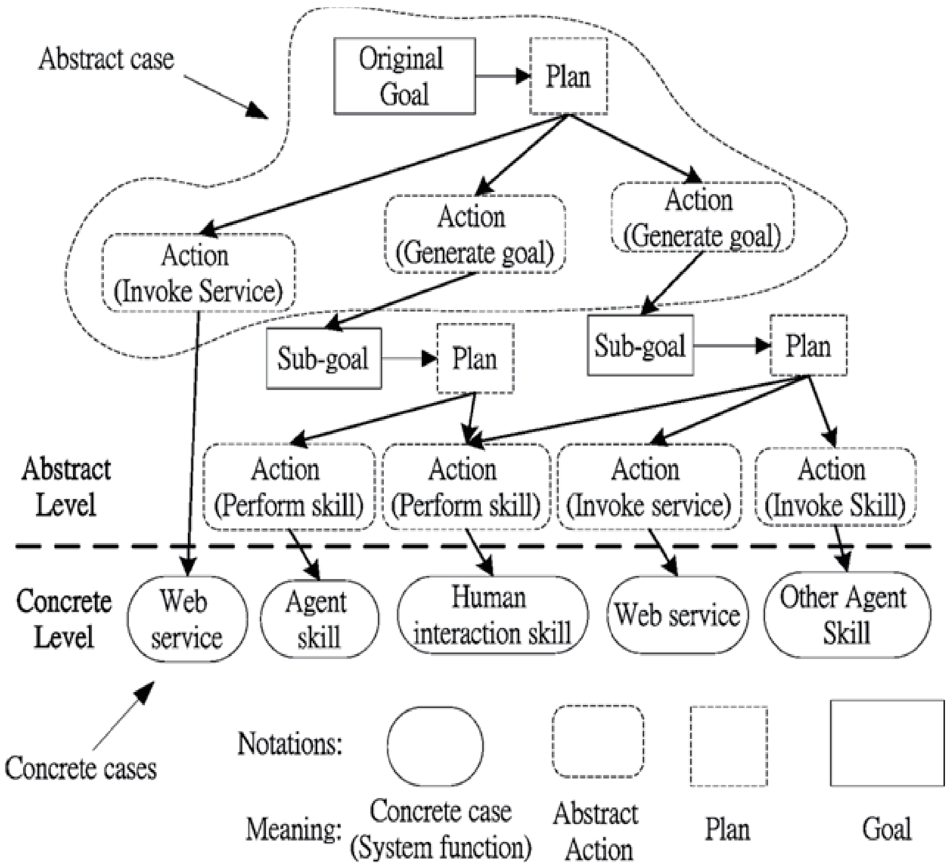


Figure 6. A goal-plan hierarchy.

been proposed in AI research and can be used to implement goal-plan hierarchy [24]. We adopted hierarchical task networks (HTN) with conditional planning for our implementation because it is suitable for hierarchical decomposition problem.

There could be several decompositions for a given abstract action in figure 6, and the decompositions are stored in the plan library for fitting different situations. Figure 7 illustrates a graph to demonstrate a possible plan decomposition of “BuyBook” goal. In this example, we assume that the original goal “BuyBook” is extended with a sub-goal “GetInexpensiveItem.” The original abstract action “BuyBook” can be decomposed into three actions “QueryBook”, “OrderBook”, and “SortItem”. By referring to the predefined domain ontology, the agent can infer that book is an item, so the action “SortItem” can be used to sort the retrieved book list to fit the sub-goal “GetInexpensiveItem.”

We assume that the original user intention BuyBook is extracted from the user’s query string and two implicit intentions QueryBook and GetInexpensiveItem are also derived. These intentions are the external preconditions of the BuyBook action.

The QueryBook and OrderBook actions are an abstract action and should be further decomposed. The actions decomposition are shown in Figure 8(a) and 8(b). There are three concrete actions which should be performed for the QueryBook action: GetService, SelectService, and InvokeService. The action GetService calls a system function to generate a query document to the service broker and get the services information from the broker. The information contained in the original goal model will be used as the basis for query document generation. The SelectService action calls a predefined system function to select a proper service for query book information if there is more than one service returned from the broker. Finally, the requester agent will invoke the service provider to request a book list for user to select the book which the user wants. The actions of OrderBook is similar to QueryBook, but because order a book need the detain book information, the requester agent has to interact with the user for getting the book.

A system architecture

A general system architecture for the intention-aware service-oriented system is shown in Figure 9. The proposed architecture gives a framework and template for the designer of the intention-aware service-oriented system. By referring the requirements specification of the system, the designer can obtain the system design which based on the proposed framework. The function of proposed architecture can be divided into three parts.

1. Personalized user intention extraction. This part implemented by Goal structure, Goal retriever, and Goal reasoner modules. Goal structure is a database which store goal models and relationships among goal models. Goal retriever implements the case-based intention extraction approach to retrieve related goal models. Goal

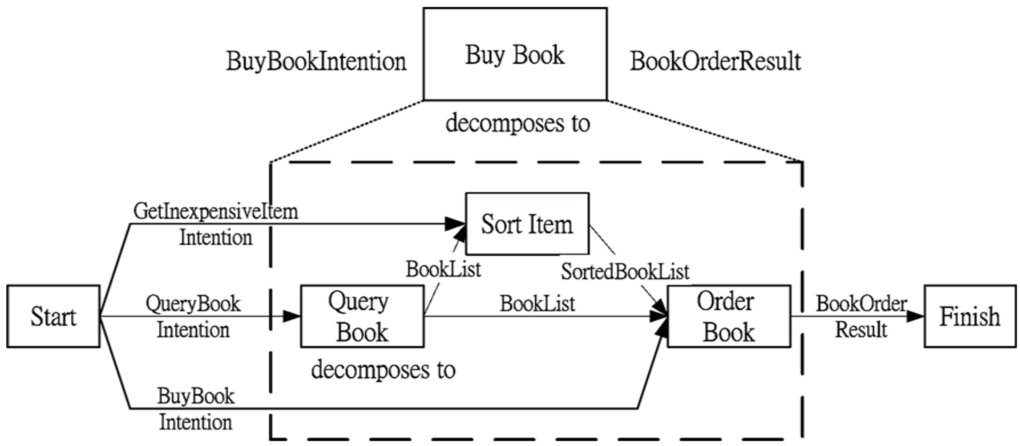
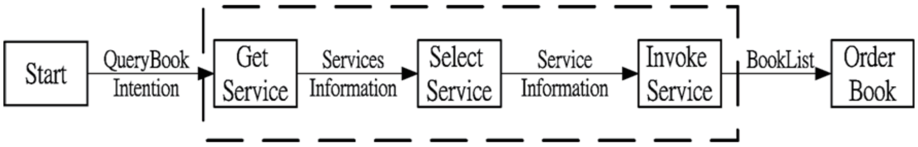
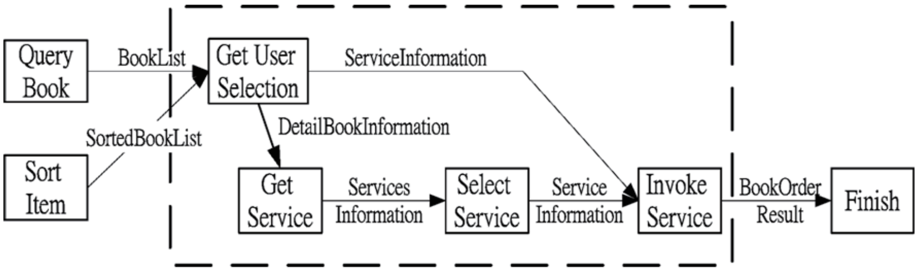


Figure 7. A plan decomposition.



(a)



(b)

Figure 8. Possible action decomposition of QueryBook and OrderBook.

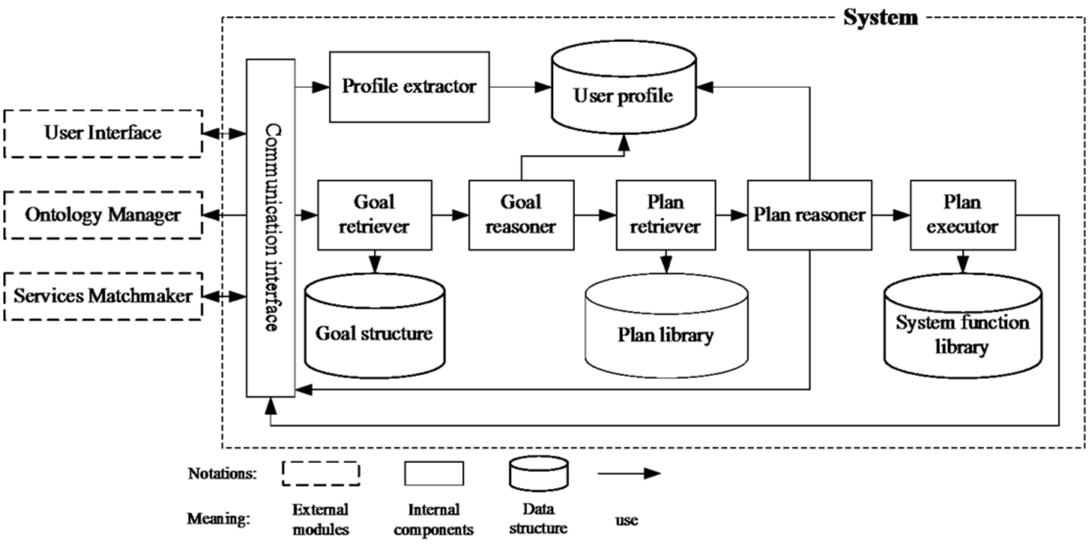


Figure 9. Architecture of an intention-aware service-oriented system

reasoner use the information in the retrieved goal models and a goal selection algorithm for selecting and adapting candidate goal models to represent the user service request intention.

2. User intention satisfaction. This part implemented by Plan library, Plan retriever, and Plan reasoner modules. Plan library contains the old goal-plan hierarchy for previous user intention satisfaction. Plan retriever use the identified goal model and the description part of the goal-plan pair in the play library to retrieve candidate plans for satisfying the newly goal model. Plan reasoner use its plan adaptation knowledge to modify the candidate plan into a new plan that could be used to satisfy the user request.

3. Plan execution. There are two modules engaged in this part: Plan executor and System functions library. Plan executor executes the plan generated in part 2. The System functions library store entries of concrete system functions. When the plan is executing, the System functions library will be used to call

concrete functions.

The Communication interface provides a set of function to communicate with the user interface and the external services matchmaker. The system also can connect with an Ontology manager such as Protégé [25] to store, query, and update the ontology. Because the main purpose of our approach is to demonstrate how to produce a suitable plan for user intention satisfaction, the plan executor in our system is only designed for testing. How to execute the plan is a complex problem because the external services exist in a nondeterministic domain. The different approaches proposed to handle the problem of planning and acting in nondeterministic domain could be found in [24].

Based on the framework of intention-aware service-oriented system, when the system designer wants to design and implement a new system, the main task of the system designer is to design and implement the Goal structure, Plan library, the System functions library, and the adaptation knowledge of Goal reasoner and Plan reasoner. Most of the other part of the system can be reused.

5. CONCLUSION

We proposed a process of how to map the service requests to the goal models and how to generate a plan to satisfy the goal models is also introduced. Two issues of designing an intention-aware service-oriented system are discussed: the intention extraction problem and the intention satisfaction problem.

The goal models are derived from the software requirement specification and the goal-driven use case requirement analysis approach is used to elicit goal models. A case-based approach is introduced for solving the intention extraction problem. The background knowledge of domain ontology, goal structures, and user profiles are used in our approach to facilitate the intention extraction process. By the learning capability of CBR, the intention extraction tool which based case-based approach can extend its intention interpretation capability. By using the proposed intention extraction approaches, the system can generate goal models to represent user's intentions by interpreting the user entered service request string. The proposed intention extraction method can be further enhanced by the research of the intention-based HCI systems. We believe that this is a useful linkage to the large body of literatures in conversational dialogue systems, multimodal interfaces, and other recognition-based systems.

Based on the generated goal model, the system can perform a series of actions to achieve the goal. We use AI planning techniques for generating these actions. Once the goal has been achieved, the user intention is satisfied. We also propose a system architecture to demonstrate how to integrate the intention extraction and intention satisfaction approach. The architecture can be used to enhance service-oriented systems with intention-aware capability. We believe that services accessing performance of service-oriented systems could be improved by adding an intention-aware mechanism into the systems and it can also facilitate the interaction between the human user and a service-oriented system.

ACKNOWLEDGEMENT

This paper was partially supported by the National Science Council in Taiwan, R.O.C through Grant NSC- 96-2218-E-235-001-MY2 and Department of Industrial Technology, Ministry of Economic Affairs, Taiwan, R.O.C. through Grant 96-EC-17-A-02-S1-029.

REFERENCES

[1] M.P. Singh and M.N. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*, Wiley, 2005.

[2] M.N. Huhns and M.P. Singh, "Service-oriented computing: key concepts and principles," *IEEE Internet Computing*, vol. 9, no. 1, 2005, pp. 75-81.

[3] J. P. Bigus and J. Bigus, *Constructing Intelligent Agents Using Java 2nd*, John Wiley & Son, New York, 2001.

[4] C. Breazeal, "Social Interactions in HRI: The Robot View," *IEEE Trans. Systems, Man*, and *Cybernetics*, May 2004, pp. 181–186.

[5] H. Wang, J.Z. Huang, Y. Qu and J. Xie, "Web services: problem and future directions," *ELSEVIER J. Web Semantics*, vol. 1, no. 3, April 2004, pp. 309-320.

[6] http://www-306.ibm.com/software/solutions/

[7] http://www-128.ibm.com/developerworks/library/specification/ws-bpel/

[8] M. Paolucci and K. Sycara, "Autonomous Semantic Web Services," *Internet Computing, IEEE* , vol. 7, no. 5, Sept.-Oct. 2003, pp. 34 – 41.

[9] http://www.w3.org/Submission/OWL-S/

[10] K. Sycara et al, "Distributed Intelligent Agents," *IEEE Expert*, Dec. 1996, pp. 36-45.

[11] K. Sycara et al., "Automated discovery, interaction and composition of Semantic Web services," *ELSEVIER J. Web Semantics*, vol. 1, no. 1, Dec. 2003, pp. 27 – 46.

[12] J.M. Vidal, P. Buhler, and C. Stahl, "Multiagent systems with workflows," *Internet Computing, IEEE*, vol. 8, no. 1, 2004, pp. 76 – 82.

[13] K. Sycara, et al., "Dynamic discovery and coordination of agent-based semantic Web services", *IEEE Internet Computing*, vol. 8, no. 3, May-Jun 2004, pp. 66 – 73.

[14] C.H.L Lee and A. Liu "Model the query intention with goals," In Proc. of the USW2005, Mar. 2005, pp. 535-540.

[15] C.H.L Lee and A. Liu "Toware intention-aware semantic Web services," In Proc. of the IEEE SCC2005, July. 2005, pp. 69-76.

[16] P. Loucopoulos and V. Karakostas, *System Requirements Engineering*, McGraw-Hill, Berkshire, UK, 1995.

[17] J. Mylopoulos, L. Chung, and E. Yu, "From object-oriented to goal-oriented requirements analysis," *Comm. of the ACM*, vol. 42, no. 1, 1999, 31-37.

[18] J. Lee, N.L. Xue and J. Y. Kuo. "Structuring Requirements Specifications with Goals". *Information and Software Technology*, vol. 43, 2001, pp. 121-135.

[19] J. Allen, *Natural language understanding 2nd ed.*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, 1994.

[20] C.D. Manning and H. Schutze, *Foundations of Statistical Natural Language Processing*, The MIT Press, Cambridge, 1999.

[21] C.H.L Lee and A. Liu, "An intention-aware interface for services access enhancement," In Proc. of AHUC2006 conjunction with SUTC2006, Taiwan, 2006.

[22] C.H. Lee and A. Liu, "Aiding User Intention Satisfaction with Case-Based Reasoning in ATIS application," In Proc. IEEE SMC2006, Taipei, Taiwan, 2006.

[23] B. Smyth, M.T. Keane, and P. Cunningham, "Hierarchical case-based reasoning integrating case-based and decompositional problem-solving techniques for plant-control software design", *IEEE Trans. on KDE*, vol. 13, no. 5, pp. 793-812, Sept-Oct 2001.

[24] S. Russel and P. Norvig, *Artificial Intelligence: A Mordern Approach 2ed.*, Prentice Hall, London, 2003.

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