A micro-task model for user-oriented semantic service description

Wei Wang 1*, Juanni Li 1 and Liang Bai 1

1School of Science, Xi’an Shiyou University, Xi’an, Shaanxi, 710065, China

*Corresponding author’s e-mail: wangwei_sci@xsyu.edu.cn

Abstract. SOA (Service-Oriented Architecture) has become a popular technology for developing network application. However, the current problem is new service cannot be quickly discovered, orchestrated and composed to match customer needs for insufficient capacity of semantic description of Web services. There is thus a need for an efficient way to describe services from users’ perspective, instead of only realization-oriented description, to make web service the most widely used. This paper proposes a user-oriented semantic service description method which combines task model with semantic service to find a novel way that is suitable and intuitive for a service consumer. A micro-task model is proposed to enhance the capacity of semantic service description from four dimensions: user goal, context, displayed variable, activity; simultaneously, a services description framework based on micro-task model is proposed to accurately map user task into services.

1. Introduction

With the arrival of ubiquitous computing age, information technologies weave themselves into the fabric of everyday life. All kinds of network technologies are used to support execution of applications. Consequently, the notion of SOA has gained wide attention within the software design and development community. Up to now, there are many enterprises that have used SOA as a “paradigm” to guide the development of applications. SOA is about bridging the gap between business and IT through a set of business-aligned IT services using a set of design principles, patterns, and techniques [1]. As a dynamically reconfigurable architecture, it can compose and orchestrate distributed resources. The interaction process in SOA includes three primary parties: a service provider, who provides some small units of functionality and their description; a service consumer, who finds out services available to meet the goal, motivation, and requirements that define the actual problem domain; service broker, who is responsible for the service registration, management, query, call, and so on. It will be treated as a collection of services, the collection includes a variety of different types of resources, such as computing resources, storage resources.

However, it is difficult to find a unified, applied method to connect the service provider and the service consumer. On the one hand, there is not necessarily a one-to-one correlation between needs and functional unit. With a different range of users’ backgrounds and the context of use, software needs are diverse, complex, and require frequent modifications. Consequently, it is hard to keep the services up-to-date. Any given need may require the combining of numerous services while any single service may address more than one need. On the other hand, since services available for building workflows are heterogeneous and self-similarity, existing Web service lacks obvious semantics to achieve service discovery, execution and combination automation for the different service consumer. The question addressing how to associate the requirements and services smoothly remains open.
2. Related works
As a partially layered architecture, the service-oriented modelling approach includes high-level goals decomposition, searching, binding and invocation services available according to an externalized service description. Some research work has been focusing on identifying the goals that service consumers want to achieve and how to find services able to reach such goals. OWL-S [2] provides an appropriate, computer-interpretable representation language framework within which these descriptions employ a standard ontology, consisting of a set of basic classes and properties, for declaring and describing services. Web Services Description Language (WSDL) [3, 4] is an XML-based interface definition language that is used for describing the functionality offered by a web service. A WSDL document uses the “service”, “port”, “operation”, “message”, and “binding” elements in the definition of network services to solve the heterogeneous description of diversified service problem. Business participants and technical participants, consumer of service and provision of service have different flavors of the service [5]. A Multi-Technologies Supported Service-Oriented Architecture (MTSSOA) is proposed by GaiHai Li to carry out semantic extension of SOA. Tuple Space is introduced as primary technology of semantic coordination [6].

Most existing work has, however, focused on complex service descriptions for automated composition. The use of services still has obvious difficulties: the information which can be obtained from the services provider is limited and technology-oriented, and the requirement of end user is abstract and user-oriented. There is no unified description method, consequently services are hard to find, and the interoperability is poor.

In order to establish seamlessly connection between requirements and services, the main challenges addressed in services description are twofold: how to create a unified description method to make the communication more smoothly between service provider and service consumer, and how to create a flexible method which provides better matching between individual needs and service functionality, consequently the value of service is improved. It needs a mechanism to help keep description simple and understandable for all stakeholders.

3. The micro-task model of user-oriented semantic service description
Tasks are activities that users have to be performed to reach a goal. Task modeling has been used at different stages of system development, from early requirements specification through to final system evaluation [7-9]. Task model describes the application in a user-oriented way and it is potentially the most powerful method available to make a supplement to current implement-oriented service description. In the SOA ecosystem, however, conventional task analysis approaches lack the ability to accurately describe the users’ need in dynamic environments. In order to seamlessly support users to develop applications that meet individual needs with services available, the main idea of user task model is applied to the process of identification, specification and realization of services. Instead of trying to represent these services from the low-level description of implementation, we use a more abstract, user-oriented task description method which can match more efficiently data information between service consumers and service providers through a description unit with small granularity. On the one hand, this method can focus on business requirements avoiding low implementation techniques; on the other hand, it can describe requirements in a more flexible way and provide cooperation mechanism and interoperability between them.

Services can be freely designed as a functional and reusable business process logic. In the past, the service entity encapsulates information only relating to publication, such as the provider organization name, the author and a text functionality description. There is a lack of the information on the use of context and the user interfaces. Consequently, it is difficult to accurately locate existing services, and moreover, this traditional description method is not suitable for the organization of small services. The concept of micro task has been suggested for the service description from multiple dimensions. Instead of a top-down, linear predefined task analysis method, the method with smaller building block and loose coupling relationship can represent dynamic tasks in a more flexible and accurate way. The idea of
micro-task description method is consistent with the feature of SOA that system constructors can be organized in a loose and flexible way.

A micro-task is a set of activities that should be performed by specified users to achieve specified goals in a specified context of use. Each instance of services can be described with a micro-task which indicates all the main scenarios or branches associated with the service. A micro-task can be a task unit associated with a specific user’s goal which is the state change derived by its performance. But micro-task is not necessarily an atomic task, it can be composed by one or more atomic tasks. For example, “login” is a task when a user enters a system, it can be decomposed into lower level tasks, such as entering user name, entering password and confirm command. “login” is a micro-task, the reason is that it is related directly to user goal access to system, whereas entering user name, entering password and confirm command are performed operations/actions for achieving the specific goal, they do not belong the set of micro-task. In a word, each instance of micro-task is an interaction process that yields an observable result of value to a particular actor.

3.1 Dimension of micro-task description
In order to more accurately describe a service, a micro-task model can be represented from four aspects: goal, displayed variables, activity and context [10]. For a good understanding of “micro-task” in this paper, we introduce the following definitions.

- **Definition 1**: Each service is represented by a micro-task \((MT)\) which can be defined as a four-tuple in a formal way: \(MT = (Goal, Displayed Variable, Action, Context)\). Where “Goal” is a desired state when the task completed. The attributes of goal include name, description, role, initial state, final state. “Displayed Variable” is a non-empty set of interface elements. “Action” is a set of actions which is closely related to the realization of the goal. “Context” includes all environmental variables associated with the target.

- **Definition 2**: Formally, the context is a combination of a finite tuple: \(Context = \{(Tem_0 \mid Spa_0 \mid Res_0 \mid Soc_0), (Tem_1 \mid Spa_1 \mid Res_1 \mid Soc_1), \ldots (Tem_n \mid Spa_n \mid Res_n \mid Soc_n)\}\). Where “Tem” is “Temporal” which includes those information of date, time, season, etc. “Spa” is “Spatial” which includes position and orientation information. “Res” is “Resources” which includes those resources available and proximity information. “Soc” is “Social relationship” which includes associated population and collaborative division to achieve the same objective.

- **Definition 3**: These concepts are related to others in specific way, there are some relationships defined in our meta model:
  - **Compare_with**: The Compare_with relationship describes the comparison function between reference signal “task goal \((r)\)” and perceptual signal “displayed variables \((p)\)”, the difference is specified as “error signal \((e)\)”. The relationships among them can be expressed: \(e = r - p\).
  - **Trigger**: If there are some difference between “task goal” and “displayed variables”, the actions will be triggered. With respect to conventional event-trigger-task methods, there are some possible trigger types such as: AND, OR, NEXT; in our method, there are one and only trigger types: IF ONLY, if only the “error signal” is non-zero, the actions would be triggered.
  - **Acting_on**: The Acting_on relationship specifies perceptual agent take some action on “displayed variables” which present on user interface to make the sensed states reaching agreement with the desirable conditions.
  - **Influence**: In dynamic environment, “task objective”, “displayed variables” and “action” would be influenced by many unpredicted factor, which is any change that affects the way in the perceptual input influences the measured output.

- **Definition 4**: The task type indicates the allocation of tasks’ performance, those tasks with similar kinds of abstraction are grouped in the same category. We inherit the Concur Task Tree(CTT) notation, introduced by Fabio Paterno [11], and extend the semantic of “CTT Task Type” according to context and context-aware application. There are four types of tasks are supported in our notation: user tasks, application tasks, abstract tasks and interaction tasks. The interaction process happens not only
between systems and users, but also between the environment information from the context of use and system.

3.2. The relationship of micro-tasks

Micro-task provides a goal-oriented description of interactive system and the relationship between micro-tasks represent the logic connection between user target units. As the operations during the execution of a task is determined by the user (user-oriented), we propose four operators to describe their relationship.

- **Definition 5**: The relationship between micro-tasks can be defined as: sequence, concurrent, replace and insert. Table 1 shows more the details about operators of micro-tasks.

Table 1. The relationship between micro-tasks.

| Relationship | Meaning |
|--------------|---------|
| sequence(seq) | When the current micro-task terminates, it enables the next micro-task. |
| concurrent(con) | Two micro-tasks can be performed in any order without any specific constrains. |
| replace(rep) | Stop the current micro-task, perform a new micro-task. |
| insert(ins) | Insert new micro-task during the execution of the current micro-task, the current task suspends, and then when the new task is terminated, the current task can be reactivated from the state reached before the interruption. |
| choice(cho) | Micro tasks can be performed in any order, once one task is chosen, other tasks are disabled. |

4. Services description framework based on micro-task

The services description framework based on micro-task is shown as figure 1. It is worth noting that there are different data formats and hierarchical structure between services, so it is needed to build an intermediate representation for the information. On the Providers side, we describe a service with a micro-task which focuses on four aspects: goal, activities, context and displayed variables. This is a supplement to the traditional description method and it represents information explicitly during the service design process. By defining system responsibilities with black-box, the aim is to map from low-level descriptions of services to the more abstract, user-oriented descriptions.

On the Consumers side, user goals are usually varied from the level of an enterprise to specific behavioral sub goal. In our method, task analysis can help each goal to find implementation (service) matching with it. Tasks can be grouped at many levels of granularity, from one or a few small steps, up to enterprise-level activities. The abstract task can be decomposed by hierarchy analysis to micro task units.

Apart from the service data, the intermediate Constructor transferred between two sides should find out these services related on consumer’s requirement from existing micro-task sets, and organize workflow according to definition, rules and constraints. A service, with self-contained and self-describing characteristic, includes description and implementation. The service can create new combinations of services according to the needs, and it can also be composed of other modular service. For each service, the description is represented in explicit way, whereas it hides the internal implementation and complexity.
5. Conclusion
The main purpose of this work is to make the communication more smoothly between service provider and service consumer through a services description framework based on micro-task. In future, we intend to continue working to research on formal description of services, development prototype of model transformation tools, which can provide a technical basis for end-user development.

Acknowledgments
The research is supported by the Scientific Research Program Funded by Shaanxi Provincial Education Department (Granted No. 16JK1602) and the Innovation and Technology Fund of Xi’an Shiyou University (Granted No. Z07074).

References
[1] Arsanjani, A. (2004) Service-Oriented Modeling and Architecture. https://www.ibm.com/dev operworks/library/ws-soa-design1/
[2] Martin, D., Burstein, M., Hobbs, J., Lassila, O., Mcdermott, D., Mcilraith, S., et al. (2004) OWL-S: Semantic Markup for Web Services. https://www.w3.org/Submiss ion/2004/SUBM-OWL-S-20041122/
[3] Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2014) Web Services Description Language (WSDL) 1.1. https://www.w3.org/TR/2001/NOTE-wsdl-20010315
[4] Yu, L., Lu, Y., Zhu, X. J., Qiu, S. W. (2014) A web services description language-based description model of internet of things services. Sensor Letters, 12(2): 448-455.
[5] Perrey, R., Lycett, M. (2003). Service-oriented architecture. In: Proceedings of the Symposium on Applications and the Internet Workshops (SAINT’ 03 Workshops). Orlando, FL. pp. 116
[6] Li G, Huang G, Yang D, et al. (2015) Study on Ontology-based Multi-Technologies Supported Service-Oriented Architecture. In: 2015 International Conference on Education Technology, Management and Humanities Science (ETMHS 2015). Tianjin. pp. 877-881.
[7] Paterno, F. (2001) Model-based design and evaluation of interactive applications. Palgrave Macmillan UK, London.
[8] Dan, D., Neville S. (2003) The Handbook of Task Analysis for Human-Computer Interaction. CRC Press.
[9] Seffah, A. (2015) A Pattern Framework for Task Modeling in Smart Environments. In: Patterns of
HCI Design and HCI Design of Patterns. Springer, Cham. pp. 81-107.

[10] Li J.N, Hua Q.Y, Ji X. (2015) PCT-DTM:PCT-Based Dynamic Task Model. Journal of Computational Information Systems, 11:2, 475–482.

[11] Paternò, F. (2002) ConcurTaskTrees: An Engineered Approach to Model-based Design of Interactive Systems. The handbook of analysis for human-computer interaction. pp. 483-500.