

# SaaS Modeling Tool for Personalization

Liu Ying<sup>1,2</sup>, Zhang Bin<sup>1</sup>, Liu Guoqi<sup>2</sup>, Wang Deshuai<sup>1</sup>, Zhang Yichuan<sup>1</sup>

<sup>1</sup>(College of Information Science and Engineering, Computer Applications, Liaoning, Shenyang, 110819)

<sup>2</sup>(Software College, Software Engineering, Liaoning, Shenyang, 110819)

{liuying, zhangbin}@mail.neu.edu.cn, liugq.neu@gmail.com, sunwonders@126.com, zhangyichuan\_33@163.com

**Abstract**— Software as a service (SaaS) is an emerging software framework satisfying many tenants. It requires a unified subscriber to describe SaaS to make for easy integration with other application; however, SaaS provides services to different tenants by running only one instance. In order to satisfy personalized needs, we propose the use of Web Services Conversation Language (WSCL) to express the views of tenant and provider separately. We also put forward a new modeling Tool for constructing SaaS Service, describing the modeling method and process. Finally we conclude the whole work in the future.

**Keywords**-SaaS Service; ex\_WSCL; Multi-Tenant; Business Model; Tenant Model

## I INTRODUCTION

Software as a Service (SaaS) is a new emerging software design, implement and deploy model<sup>[1]</sup>. SaaS applications contain business data and logic which are usually required to integrate with other applications<sup>[2]</sup>. It needs a unified subscriber to describe SaaS to make its integration easy. The service provider delivers software functionalities with one single instance software application running for all of its tenants<sup>[3]</sup>. This one instance is used by different tenants having different personalized needs in terms of data, process rules, and business rules. Modeling a SaaS application based on service description is difficult when multiple parties are interacting. WSCL provides a conversation to conveniently describe the coordination between SaaS provider and tenants. However, when a business model is described from the provider's view, it must define complex business logic including heterogeneous data formats, business rules, and business processes. WSCL restricts this by only allowing the definition of one group of Input & Output documents in an interaction -- not allowing processing of heterogeneous data. In addition, the standard definition of transition defines the condition for source interactions through a data format, and it

does not consider heterogeneous business rules in the condition. In this work, we extend WSCL to satisfy the needs above. The SaaS application provider can use the extended WSCL (ex\_WSCL) to handle the business model from his view. We also build a modeling tool as an interface to simplify the modeling process. The business model is meant to be implemented to run in the server of the SaaS application provider, and different tenant models form a subset of the business model. To assist the tenant, we build an algorithm to create a tenant model from the business model automatically in the modeling tool.

## II SAAS SERVICE MODEL BASED ON EX\_WSCL

WSCL allows each involved participant in the conversation to express his business logic himself, so it is suitable for modeling a SaaS Service. However, WSCL has restrictions about expressing heterogeneous data formats, business rules, and business processes in one flow. In the definition of WSCL, conversations contain interaction and transition. Interaction models the actions of the conversation as document exchanges between two participants. One interaction maps to one document exchange group, and supports five types, which are Receive, Send, ReceiveSend, SendReceive, and Empty. As the SaaS application is required to run one instance to fit all of its tenants, a one-to-one mapping in interaction can not satisfy the needs of heterogeneous data formats. Transition specifies the ordering relationships between interactions. A transition specifies a source interaction, a destination interaction, and, optionally, a document type of the source interaction called SourceInteractionCondition as an additional condition for the transition. Only document type restriction in transition is not sufficient, so it must add flexible definition of business rules.

One Interaction does not map to one group of information exchange, but to several groups. This means that one interaction could bind with several groups of input and output.

One transition only contains one SourceInteraction and one DestinationInteraction, but would not be suitable for defining

concurrent structured transitions. In concurrent structured transitions, the exchange document of several SourceInteractions must all satisfy the constraints and the transition triggers for the DestinationInteraction to be executed. We therefore extend the definition of transition to allow it to have several SourceInteractions and only one DestinationInteraction. We also modify the transition condition SourceInteraction, and allow it not only to contain elements of the exchange document but also their compound expressions.

When triggering a transition in a business process, there are business rules for its precondition except the document type restriction in source interactions. We add business rules in the definition of transition and require them to be part of compound logic expressions to constrain attributes in the interaction's I/O.

### III SAAS SERVICE MODELING

In this paper, SaaS service model based on extended WSCL uses conversation as basic element for describing its business logic. When constructing a SaaS application based on Web Service, the provider could construct a main process firstly, and then describe it as Business Model by extended WSCL. When a tenant wishes to subscribe to the SaaS application, he could choose the interactions and a sequence of them included in business model. According to his choice, the provider separates their jointly owned interactions from the business model, inverts the operations in them, and produces a tenant model. The tenant gets his model from his view, and does not need to know all of the complex business logic in the business model. If he wants more functionality or a different business flow, he could send his personalized requirement to the provider. The provider may modify his business model by adding new interactions or transitions or new interface definitions, and publish a new model.

For convenience to both the provider and the tenants, we provide a modeling tool. We can use it to create an effective state diagram and to document WSCL for business service. The tool implements two types of functional modules, which are SaaS Service state diagram modeling and WSCL document conversion with state diagrams. The first functional model is implemented with graphical editor framework (GEF) and rich client platform (RCP). The second functional model utilizes reverse engineering to transform WSCL into a state diagram

and vice versa. The interface for the modeling tool is shown in Figure 6.

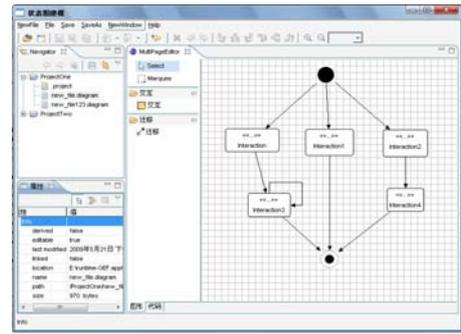


Figure 1. Modeling Tool for SaaS Service

In addition, the tool provides for obtaining the tenant model from the service model according to the tenant's personalized policy automatically. Firstly, we transform the WSCL document of the service model to a graph, as described by the work of Daniela et al<sup>[4]</sup>. Secondly, as we focus on different processes for customization in this paper, we describe how to implement to choose one sub-process from several ones and delete the unselected branches in the process.

### IV CONCLUSIONS

This paper proposes a modeling method based on Web services for SaaS applications with multi-tenancy support. We have constructed a demonstration version to run the SaaS application in this way. Adding running information in WSCL is the goal of our future work.

### V ACKNOWLEDGEMENT

This research is supported by “National Hi-tech Research and Development Project (863) of China” (2009AA01Z122), “National Natural Science Foundation of China under grants” (60903009, 61073062) and “Natural Science Foundation of LiaoNing Province” (20102061).

### REFERENCES

- [1] Iod, Software as a Service, Kogan Page Ltd, 2002.
- [2] Sun Wei, Kuo Zhang, Shyh-Kwei Chen, etc. “Software as a Service: An Integration Perspective”, ICSSOC 2007, pp 558-569.
- [3] Thomas Kwok, Thao Nguyen and Linh Lam.A. “Software as a Service with Multi-tenancy Support for an Electronic Contract Management Application”, 2008 IEEE International Conference on Services Computing, pp 179-186.
- [4] Daniela G, Juan C C, Mokrane B. Behavioral matchmaking for service retrieval. ICWS '06. International Conference on Sept, 2006 ,145-152