Service Prototyping for Service Testing in Virtual Reality

Kyuhyup Oh, Jin Sung Lee, Sang-Kuk Kim, Jae-Yoon Jung, and Bohyun Kim

***Abstract*—As the service sector rapidly grows, service prototyping and testing become more important. The main purpose of this research is to develop the service prototyping technology for service testing in virtual reality. The service prototype is the model that abstracts the core characteristics of real-world service and it can be used to demonstrate the real-world service in virtual reality. To do this, we construct the service testing system that shows the service scene upon service process by service execution system interworking with database. This system provides flexible and scalable test environment by separate virtual reality implementation and service scenario. In addition, we describe the case of self-service car-sales space design applied the technology.**

***Index Terms*— Service prototyping, virtual Reality**

I. Introduction

As the service sector rapidly grows, the service prototyping and testing become more important to perform successfully new service development and service innovation [1]. Unlike typical products, service prototyping and testing are difficult due to the characteristics of inherited service such as invisibility and intangibility [2]. Recently, as information technology is widely developed, a few advanced service test methods are introduced. For example, ServLab, developed by the Fraunhofer Institute, provides the service test laboratory, where the service prototype is developed in virtual reality to test the real-world service of a client, various interactions of service providers and customers can be experimented to innovate the service process or design the new service [3]. In addition, the Korea Institute of Industrial Technology (KITECH) is also developing a service test laboratory, named s-Scape, for service design, test, and analysis in virtual reality. s-Scape supports experiment environments where service consultants can represent various service scenarios in virtual reality to design the new service or improve the existing service [4].

In this paper, we described a service prototyping technology to prototype and test real-world services such as a car-sales shop and a duty-free shop in virtual reality. Specifically, for the purpose of automatically executing the service prototype in virtual reality for the service testing, the service prototype diagram (SPD) was introduced as a service prototype design tool and the service prototype description language (SPDL) as an executable service description language [5], [6]. SPD diagram and SPDL language help flexible and rapid service prototyping by separating the implementation of virtual reality contents and the modeling of service test scenarios. Finally, we illustrated the service prototyping with the case study of new service design for self-service car-sales.

In the rest of this paper, we described the overview of our service test laboratory, s-Scape, in Section II. The main tools for service prototyping, SPD and SPDL, were introduced in Section III, and the execution system for the service prototypes was explained in Section IV. The case study was then presented for the verification of the system in Section V. Finally, we concluded this paper in Section VI.

Manuscript received October 12, 2012; revised November 25, 2012. This work was supported by the Industrial Strategic Technology Development Program of MKE/KEIT [10035122-2011-02, Construction of the innovative service laboratory for service testing.

Kyuhyup Oh, Jin Sung Lee, Sang-Kuk Kim, and Jae-Yoon Jung are with the Department of Industrial and Management Systems Engineering, Kyung Hee University, Yongin, Republic of Korea (e-mails: {k8383, jinsl127, sangkkim, jyjung}@khu.ac.kr).

Bohyun Kim is with the Digital Collaboration Center, Korea Institute of Industrial Technology, Ansan, Republic of Korea (e-mail: bhkim@kitech.re.kr).

II. S-Scape: Service Test Laboratory

A. Overview

The service test laboratory, s-Scape, is the experiment environment in which we can prototype and test new or existing services in virtual reality for the purpose of analysis and improvement of real-life services. The service consultants can utilize the laboratory as a service improvement and innovation tool by observing the test progress and evaluating the test results.

To effectively prototype and test real-life services, we developed the s-Scape methodology, which is composed of five phases as illustrated in Figure 1. We first define the service test project by identifying the customer needs for service testing. We then design the service experiment and develop the service prototype. Finally, we perform the service experiment in virtual reality and analyze the result of the experiment *s*-Scape provides a physical environment where a target service is prototyped and tested through the s-Scape methodology. As shown in Figure 2, the physical environment consists of five facilities: the system equipment room that is installed with H/W and S/W, the 3D screen that displays service scenes in virtual reality, the controller system for s-Scape systems and VR devices, the stage where participants experience the services, and the observation and discussion space for service stake-holders.

Fig. 1. Five phases of s-scape methodology

B. Types of Service Tests

We can perform two types of service testing in s-Scape according to the degree of freedom of the customer who join the service experiment in virtual reality. The first type is the free-path experiment in which customers can control their

DOI: 10.7763/IJIEE.2013.V3.323

moves among ready-made scenes and their viewpoints in virtual space by themselves. Since the customer can directly control the service space, this type of experiment is suitable to assess the mood and friendliness of physical evidence, for instance, interior design and building structure. Such experiments are also useful to analyze the service environment for individual customers. The second type of service testing is the pre-defined path experiments where experimental customers can move only in the paths that the service designer has already defined in advance for the specific service scenarios and has been stored in the database of the laboratory. As the customers choose one of the possible pre-defined paths and experience the corresponding service environments, these experiments can be effectively applied to evaluate service process by measuring significant indicators such as cost and time and service concept demonstration in new service development.

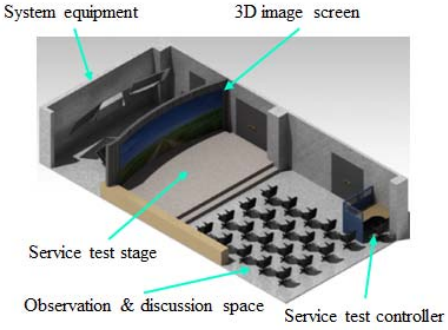

Fig. 2. Blueprint of service testing space

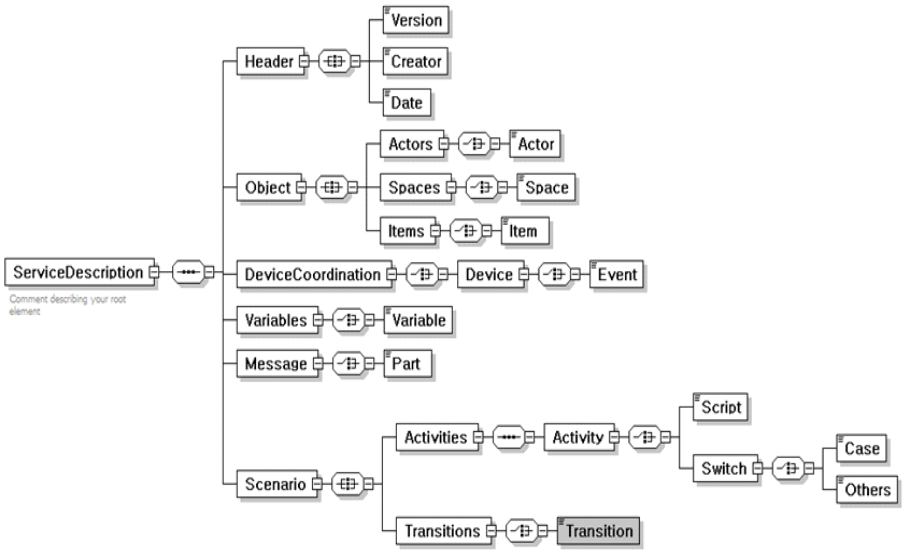

Fig. 4. XML Schema for SPDL

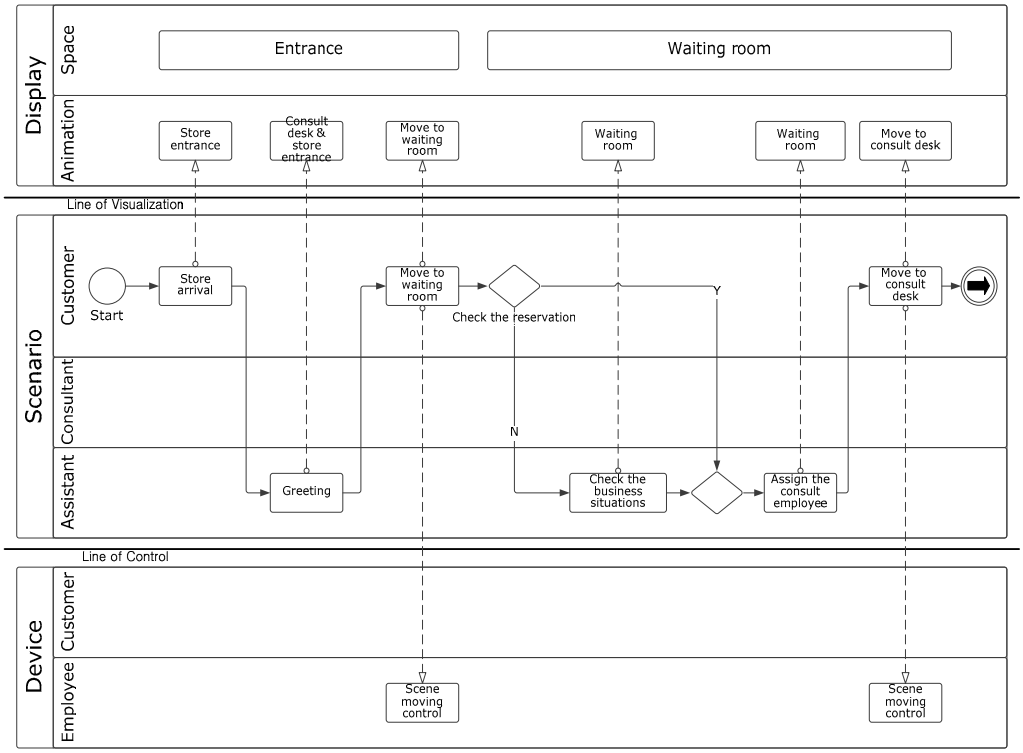

Fig. 3. Sample of SPD in the car-sales service

III. Tools of Service Prototyping

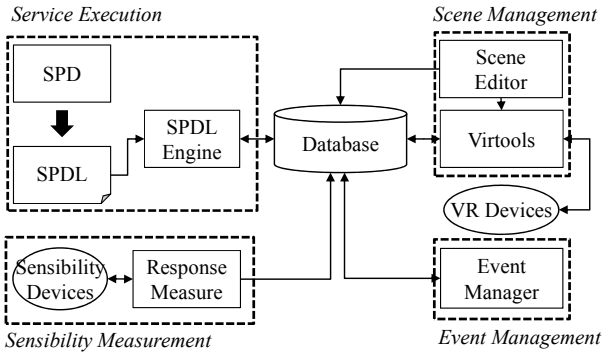

Fig. 5. System architecture for s-Scape

SPD is a service modeling tool by which service designers can draw and express service scenarios and they also set the interaction with virtual reality devices for interactive service testing in s-Scape. The diagram is devised by combining service blueprint and BPMN. Service blueprint is widely used in service design and analysis, and BPMN is a business process modeling standard providing a variety of notations. We replace the main components of service blueprint with three core components of service prototyping: display, scenario and device. The example of SPD is presented in Fig. 3 [5], [6].

SPDL is an XML language of describing a service model which can be executed in the s-Scape system, which will be introduced in Section IV. SPDL reflect the main components of SPD such as service process, service scene, and control devices. The XML schema of SPDL is shown in Fig. 5.

IV. Service Execution System for s-Scape

A. System Architecture

The software system in the s-Scape laboratory supports four functions: service execution, scene management, event management and sensibility measurement. The system components for the functions interact with one another via the shared database. The overall system architecture is depicted in Fig. 5.

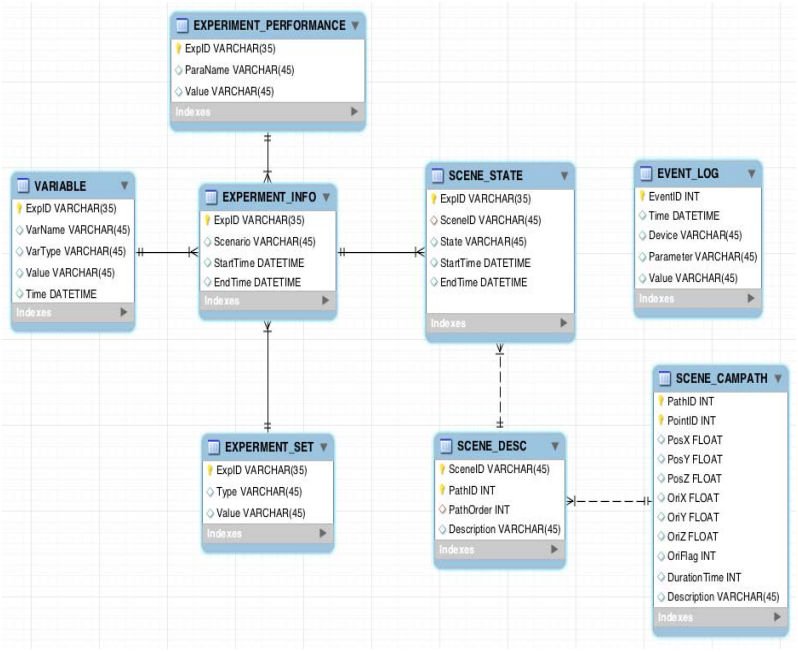

Fig. 6. Database Schema for s-Scape system

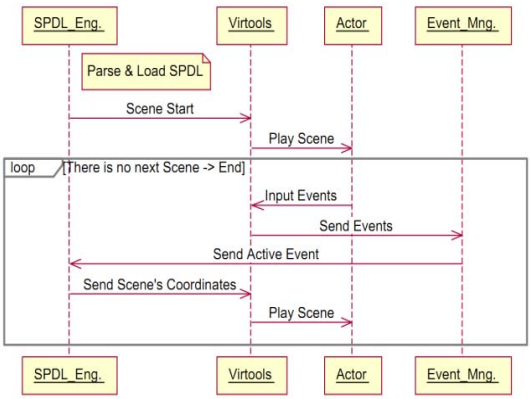

Fig. 7. Sequence diagram for SPDL Engine

1) Service execution

SPD is created to test the real-world service in virtual reality environment, and it is then converted to an XML file conformant to SPDL. The XML file is executed by SPDL engine, which replays each scene by refereeing the service scenario described in the file. The engine requests corresponding scenes to display 3D contents stored in database. In our research, 3D contents are created with 3D software, Virtools, and they are rendered in the 3D screen as introduced in Section II.

2) Scene management

Service scenes displayed in the 3D screen are viewed with 3D glasses to the participants of experiment in progress of service testing. Service scene can be controlled with camera coordinate and the point of view in service virtual space. They provides the functions such as editing the service scenes and showing the service scenes requested from SPDL engine.

3) Event management

Event Manager gathers the events of virtual reality devices and analyzes the events, and then determines the proper systematic actions related to service scenes. The events and actions have been pre-defined and stored in database, too.

4) Sensibility measurement

The system can measure brain wave, heartbeat, eye tracking of participants by using several augmented reality devices of sensibility responses during service testing. We can analyze the sensibility response data which are stored in database during the experiments.

B. Database

The database schema of s-Scape system is presented in Fig. 7. SCENE_DESC, SCENE_CAMPATH and SCENE_STATE tables are related with one another to perform the execution of service. SCENE_DESC table is used to store information about service activities in service scene. Each of the service scene has the SCENE_ID. It can have circulation of cam paths in service space and viewpoints which are stored in SCENE_CAMPATH table. The start and end condition of every executed service scenes is stored in SCENE_STATE table.

C. SPDL Engine

SPDL engine plays a role of service execution. The engine reads service process information and service scene's SCENE_ID in SPDL documents. After that, it sends service scene's data to the virtual reality engine, Virtools. The Virtools engine references SCENE_DESC and SCENE_CAMPATH tables to play the requested service. In playing the scene, the Virtuools updates SCENE_STATE table to notify service playing completion. In some decision points which have been described in a service scenario, the user can decide the next service scene. On the user's decision, the Event Manager analyzes a VR machine's event data and stored in VARIABLE table. The SPDL engine references VARIABLE table and requests to play the chosen scene in Virtools. Fig. 7 shows a sequence diagram of the execution of SPDL engine and other systems.

Fig. 8. Service prototype for testing used-car service shop

V. Case Study

In this section, a case study of service prototyping and testing in a Korean used-car sales service is introduced. This study aims to suggest a service model of self-service car sales environments and validates the model though service prototyping and testing. The scenario of the service prototype contains the following scenes:

1) A customer enters the used-car sales shop.
2) The customer walks towards cars of interest.
3) He looks around the car and tries to ride it.
4) The consultant talks to the customer for car sales when he steps down from the car.
5) The customer decides to buy the car or just go.
6) He leaves the shop.

Before executing the service prototype, we need to prepare steps as follows:

1) Define which service scenario is tested in s-Scape.
2) Design a few service processes based on the scenario.

3) Develop service scenes which are displayed to customers during a service process.
4) Generating SPDL to describe coordination with VR devices

After the target service which is displayed during service testing is designed in 3D models, Scene Editor stores the information related to each service scene for system operation in database. The information includes customers' traveling paths and coordinates of the customers' viewpoints.

SPDL engine reads service scene information for related service activities in SPDL. Figure 8 shows some screenshot and diagram of service scenario of the example service prototype, which represents service processes, service activities, and service scenes. VR engine displays 3D service scene models in 3D screen automatically or semi-automatically in response to the action events generated by participants. Event Manager receives event information about test participants' behaviors monitored in VR space. The Event Manager conducts tasks of interpreting contextual meaning of test participants' behavior events dynamically, and therefore SPDL engine can refer to the interpreted data to decide the sequence of service activity flows agilely.

VI. Conclusion

As the service industries are growing, companies are making an effort for developing new services and strengthening existing services to obtain their competitive advantages. In this paper, we introduced the service prototype technology in the service test laboratory. We presented the service prototype tools, SPD and SPDL. Finally, we illustrated a case study of service experiments with a self-service car sales scenario.

References

[1] K. J. Kim and Y. S. Hong *et al.*, "Services Innovation Research Framework and Research Issues," *Journal of the Korean Institute of Industrial Engineers*, vol. 35, no. 4, pp. 226-247, 2009.
[2] P. Kotler and K. L. Keller, *Marketing Management*, 13th ed, Pearson Prentice Hall, 2008.
[3] T. Meiren and T.Burger, "A Testing of service Concepts," *The Service Industries Journal*, vol. 30, no. 4, pp. 621-632, 2010.
[4] B.-H. Kim and K.-J. Kim, "Introduction to the innovative service laboratory for service testing," in *Proc. of Conference on Korean Institute of Industrial Engineers, 2010*.
[5] J. S. Lee, H. J. Kim, S. K. Kim, B. H. Kim, and J.-Y. Jung, "Service Modeling for Service Testing Laboratory," in *Proc. of 41st International Conference on Computers & Industrial Engineering*, 2011.
[6] J. S. Lee, K. H. Oh, C. Y. Park, S. K. Kim, B. H. Kim, and J. Y. Jung, "Service Prototype Description Language for Virtual Service Laboratory," *Journal of the Korean Operations Research and Management Science Society*, vol. 36, no 4, pp. 91-107, 2011.

Kyu-Hyup OH received BS from the Department of Industrial and Management Systems Engineering at Kyung Hee University, Yongin, Rep. of Korea, in 2009. He is currently a MS-PhD candidate in the same department. His research interests include business process management, process mining, big data analysis, and IT service.

Jin-Sung Lee received BS and MS from the Department of Industrial and Management Systems Engineering at Kyung Hee University, Yongin, Rep. of Korea, in 2007 and 2010, respectively. He is currently a PhD candidate in the same department. His research interests include business process management, IT service, and service science.

Sang-Kuk Kim received PhD in MIS from the University of Wisconsin-Madison. He is a professor in the Department of Industrial and Management Systems Engineering at Kyung Hee University, Rep. of Korea. He served as a chairman of Korean Society of MIS, and also a chairman of Korean AI Society.

Jae-Yoon Jung received BS, MS, and PhD from the Department of Industrial Engineering at Seoul National University, Seoul, Rep. of Korea, in 1999, 2001, and 2005, respectively. He is an assistant professor in the Department of Industrial and Management Systems Engineering at Kyung Hee University, Seoul, Rep. of Korea. His research interests include business process management Internet business, and service engineering.

Bo-Hyun Kim received MS and PhD from the Department of Industrial Engineer at Korea Advanced Institute of Science and Technology. He is a principal researcher in the Research Group of IT-converged Manufacturing System at Korea Institute of Industrial Technology. His research interests include CAD/CAM, virtual manufacturing system, manufacturing information system, and manufacturing-oriented service.