

An User-driven Service Creation Architecture in Consumer Networking Environments

Yuchul Jung*, Jin-Young Kim**, Hyejin Lee***, Kwang-Young Kim****, Dongjun Suh*****

Abstract

In a Web 2.0 context, users are exposed to numerous smart devices and services that allow real-time interaction between users (or consumers) and developers (or producers). For the provisioning of new user-created services based on user's context, the data management of service creation experiences becomes a non-trivial task. This article introduces a data model for service creation and then proposes a service creation management architecture which enables new service creation using the data model, the management of the service creation data, and the semantic service discovery across internal/external service repositories. The article also explains the use of the proposed architecture with two different scenarios: home and mobile environments. The proposed architecture for service creation data management offers consistent and seamless handling of the service creation data throughout its usage lifecycle.

Keywords : Web 2.0, User-driven Service, Service Creation Management Architecture, Semantic Service Discovery, Consumer Networking Environments

소비자 네트워킹 환경에서의 사용자 주도 서비스의 효율적 생성

정유철*, 김진영**, 이혜진***, 김광영****, 서동준*****

요약

Web 2.0 환경에서, 사용자는 사용자 (또는 소비자)와 개발자 (또는 제작자) 간의 실시간 상호 작용을 허용하는 수많은 스마트 장치 및 서비스에 노출되고 사용하게 된다. 사용자 컨텍스트 기반의 새로운 사용자 생성 서비스를 제공하기 위해서 서비스 생성 경험의 관리는 매우 중요하다.

본 논문에서는 서비스 작성을 위한 확장된 데이터 모델을 소개하고, 이 데이터 모델에 기반한 신규 서비스 생성, 서비스 생성 데이터의 관리, 그리고 내/외부 서비스 저장소에 대한 의미적 검색을 가능하게 하는 서비스 생성 관리 아키텍처를 제안한다. 또한, 홈 환경과 모바일 환경에 걸친 두 가지 시나리오를 사용하여 제안 아키텍처를 사용하는 방법에 대해서도 설명하며, 서비스 생성 데이터 관리를 위해 제안한 아키텍처는 서비스 생명주기 동안의 서비스 생성 데이터를 일관성 있게 관리할 수 있다

키워드 : 웹 2.0, 사용자 주도, 서비스 생성, 의미적 검색, 서비스 생성 관리 아키텍처, 소비자 네트워킹

1. Introduction

※ Corresponding Author : Dongjun Suh

Received : December 10, 2016

Revised : December 28, 2016

Accepted : Decembe 30, 2016

* Senior Researcher, Korea Institute of Science and Technology Information (KISTI)

** Researcher, Korea Institute of Science and Technology Information (KISTI)

*** Senior Researcher, Korea Institute of Science and Technology Information (KISTI)

**** Senior Researcher, Korea Institute of Science

With the advances in smart devices, such as smart mobile phone, tablet pc, smart TV etc., users more frequently communicate with other people (or other devices) through the

and Technology Information (KISTI)

***** Senior Researcher, Korea Institute of Science and Technology Information (KISTI)

Tel: +82-42-869-0671 , Fax: +82-42-869-1767

email: djsuh@kisti.re.kr

devices. In the sense of direct/indirect participation and social involvement, the convergence of the mobile communication, broadcasting, and other 3rd party Internet services signifies that users' roles are extended to creators of various types of contents as well as new services.

Within the concept of user-centric service creation, a paradigm shift on service creation environment (SCE) occurs because the service creators are not official service providers or professionals. Furthermore, these service creators (i.e., end-users) often have little knowledge of not only computer technologies, but also available web services. As traditional SCE interfaces are difficult to handle for average users, they may result in unnecessary and excessive interaction between the SCE interface and the end-users. To mitigate the disturbance, the entire creation process needs to be supported via intuitive graphical wizards or assistants.

One of the key reasons why the paradigm shift of SCE arises is that the users' needs are dynamic and spontaneous. Most of the users need can rarely be predicted well in advance. To enable more efficient service creation, current approaches guide users at any level through an SCE with intuitive user interface like Yahoo Pipes and OPUCE [1]. In addition, the difficulties can be minimized with the help of knowledge component by assisting users to find relevant Open APIs that satisfy input/output constraints with functional semantics. Although employing other users' service composition experience is the most natural way, most SCEs do not consider it yet.

More recently, [2] discussed the knowledge support for SCE. Especially, the work considered, social networking within a user group - more interactive participation & collaboration among end-users (or groups), based on the sharing characteristics of their

proposed data model which includes user and user group attributes that the Semantically Interlinked Online Community SIOC (SIOC) and the Friend on a Friend (FOAF) share simultaneously. We assume that the explicit sharing of service creation experience will accelerate the development of ecosystem of user-driven service creation. Therefore, data related to the service creation need to be managed and utilized in a more decent manner.

In this context, a key challenge for stabilizing the user-driven service creation ecosystem is how to model, share, manage, and discover the user-driven service creation data. It is noteworthy that in this article we aim to explain the user-driven service creation which utilizes the data model for service creation; illustrating how the data model can be seamlessly integrated to create a new composite service in a personalized manner; and discussing their extensibilities with a social ontology, FOAF. In addition, detailed service match-making procedures which can be arisen during the recommendation of relevant service snippets are discussed.

2. Trends, Technologies and New Challenge

Thanks to the Web 2.0 paradigm, the user-generated contents (UGC) comprising multimedia, knowledge, software, and blogs were circulated through the Internet. Together with the advances of the Service Oriented Architecture (SOA) paradigm [3], the concept of user-generated services is unleashed as an extension of the UGC in the web service domain. By adhering to the SOA principles, heterogeneous applications built with different technologies and hosted in very different machines can be owned and managed by completely independent organizations. A good

example is service mashup. A mashup is a simple and small web application made by linking the results of the computation of several services available on the Web.

Ordinary end users who have little computing knowledge & skills can not only generate contents, but also create applications or services by using intuitive service creation environments (SCEs). Yahoo Pipes and OPUCE are good examples. They provide graphical tools which allow users to make service flows (i.e., logic) using an abstract representation of atomic services and corresponding execution environments for the validation of newly generated service. For example, available web services such as Google Maps, Google Search, and Twitter Messaging are represented as building blocks with inputs and outputs. Users can build their own services by linking those blocks.

However, considering recent trends – consumers and creators build their echo-system in content generation in consumer networking, we should look at the service creation cases with a view point of new opportunities. Presuming that the number of user-generated services is going to grow in the near future, the problems of how to represent the service creation experiences and how to share them efficiently will be more crucial. In this article, we regard service mashup and service composition as the service creation experiences in this article because they combine more than one atomic services.

3. Requirement and Related Work

This section starts with defining the requirements. The three categories of related work we have witnessed, graphical SCE, knowledge base management, and semantic service discovery, are then reviewed.

A semantic search interface that supports semantic service discovery should follow existing standard technologies. In addition, the heterogeneity of information sources also must be considered. One solution is to use a common representation format such as XML and RDF, thereby providing service providers with a single interface to access all the data. This will allow the user to navigate others' prior service creation cases by providing a set of relevant service creation cases according to the user context.

Graphical SCE: To the best of our knowledge, there is no comprehensive solution (i.e., a solution that allows case base building and semantic service discovery) that can meet the all our requirement. However, there exist interesting researches in the area of graphical SCE, case base construction, and semantic service discovery. Existing SCEs follow mainly the flowchart concept [4], and more recently GUI aspects are strengthened by the Widget concept [5]. Even with continuous efforts to support intuitive and, advanced GUIs for service creation, ordinary users still have difficulties in understanding flowchart manipulation (mapping inputs/outputs, condition, loop, etc.) and manual composition steps. To overcome the bottleneck, [6] employed a goal-pattern library which maintains procedural steps for high-level user goals (e.g., “send email” and “make resume”) to speed up service creation using existing web service APIs.

Knowledge base for SCE: In terms of knowledge base, much work has been done in artificial intelligence and ontology building area. However, KB construction for user-driven SCE has not been attempted vigorously. [7] described the possible uses of how-to articles based procedural knowledge in semantic web service composition. Moreover, [2] discussed about the aggregation of service creation cases from users and the Web as

forms of collective intelligence.

Semantic Service Discovery: There exist various solutions for semantic service discovery. One major stream is to follow existing standards such as DARPA Agent Markup Language for Services (DAML-S), Web Service Modeling Ontology (WSMO), and Web Service Description Language (WSDL). However, it was not spread out between any other common users because annotating the collection of services based on those standards requires much effort and expert-level knowledge. Among other approaches, information retrieval (IR) technique-based approach is our concern. [8] employed a preference degree for web services by considering service relevance and service importance. The service relevance is term frequency (TF) * inverse document frequency (IDF) weighting based relevance measure. The latter is a connectivity measure between two web service operations based on schema matching. This shows that IR-based technique can be successfully adopted in web service discovery and ranking.

4. Data Model for Service Creation

For efficient circulation of experiences on service composition (or mashup), we've designed a particular ontology model. Fig. 1 is our proposed ontology model for service creation. The Resource Description Framework (RDF) is used to describe the used service data to make it machine-readable and interchangeable. For the querying and reasoning process upon the data, SPARQL Protocol and RDF Query Language (SPARQL) can be used because it includes required query processing functions such as conjunctions, disjunctions and optional patterns. The data model is composed of three parts: service

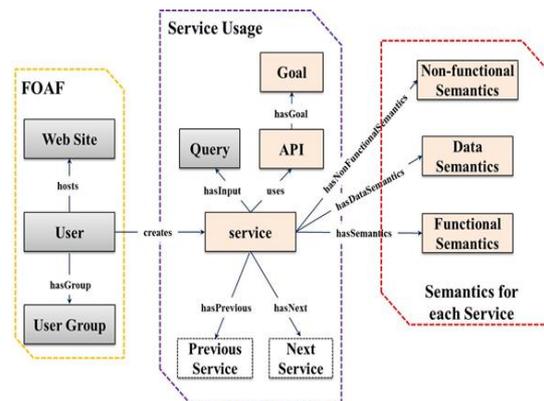
usage, semantics for each service, and social ontology connection.

The service usages part contains which service (i.e., API) is used in the previous/current/next step under specific goals (e.g., send message, send email, and obtain map). Semantics for each service includes almost all kinds of service metadata as the name of functional, non-functional, and data semantics. <Table 1> shows examples of each semantic type.

Types	Properties
Functional Semantics	service name, category, provider, and description
Non-functional Semantics	ratings, cost, communication protocol, programming language, etc.
Data Semantics	types and values of input/output parameters of the service

<Table 1> Types of Semantics

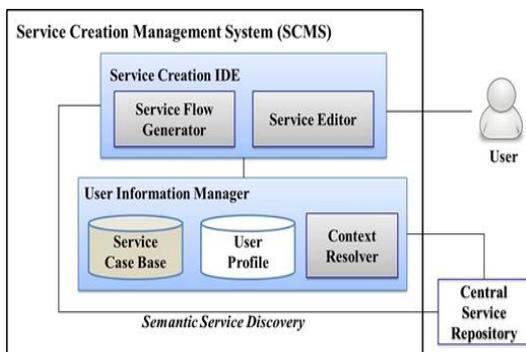
In addition, as an avenue to make the data sharable and linkable by following semantic web standards, it can be connected with other ontologies such as FOAF. [2] shows a possible relationship when service creation meets the social ontology.



(Figure 1) Data Model of Service Creation

5. Overall System Architecture

In this section, we introduce a high-level architecture of service creation management system (SCMS) which can be used for extending currently existing SCE. The high-level architecture consists of three main components, the service composition IDE, user information manager, and semantic service discovery as in Fig. 2. Central service repository (CSR) supports the functions of SCMS of each user. The CSP is running on another computing machine and it will serve a number of different SCMS.



(Figure 2) High-level Architecture of Service Creation Management System

5.1 Service Creation IDE

Users can use the service editor and the service flow generator to create a new service creation. The output is a RDF file describing the used Open APIs, the relations between required attributes. Service flow controller is used to create/update links between different Open APIs. By acting on the different links, the user can customize the composite service [9]. Service editor is used to add/revise related properties and their property values (e.g., service creator, input/output parameters, descriptions, etc.) To maximize the usefulness of available Open APIs and to ease their composition, an intuitive user friendly Web

application that displays services as building blocks to be connected in a workflow by using a simple GUI can be considered.

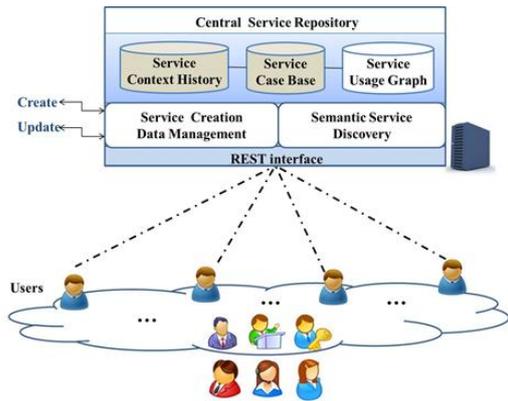
5.2 User Information Manager (UIM)

The other top-level component in Fig. 2 is the user information manager. It supports various types of user requests performed in the service composition IDE. Basically it maintains user profile (e.g., personal information including service preferences) and manages context information (e.g., user context that can be obtained from user's smart devices) so the service creation task can be done whenever needed. Furthermore, the user information manager comprises a database called "Service Case Base" which is used to store the service composition cases that are built by the SCE users.

When the user completes the service creation including slight modifications, details of the creation represented following above mentioned data model are stored and updated in the user's service case base. Upon the updates, the user's UIM module invokes a REST API for the data updates of central service repository (in Fig. 3). The REST API updates the service case base in the central service repository.

5.3 User Information Manager (UIM)

Central service repository enables users to share service creation experiences more efficiently as Universal Description, Discovery and Integration (UDDI) architecture. Overall functions are similar with ones of the UDDI, our CSR has more extended capabilities in constructing service usage graph and semantic service discovery. In terms of CRS's architecture, although a distributed architecture gives benefits in scalability, we choose a centralized approach due to its simplicity and real-time speed in updates.

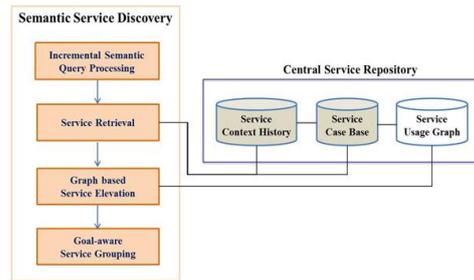


(Figure 3) Overall Architecture of Central ServiceRepository

In addition, for easy data management including creation and update tasks, and realization of semantic service discovery, the central service repository provides REST APIs.

5.4 Semantic Service Discovery

Simply put, semantic service discovery is to find context-aware service match-making across internal/external service repositories. Its match-making algorithm goes beyond the scope of simple keyword search because it deals with various types of semantics. Basically, we consider semantic factors such as functional semantics (e.g., service name, category, provider, and description), non-functional semantics (e.g., ratings, cost, and protocol, programming language.), and data semantics (e.g., input and output parameters of the service). Fig. 4 illustrates our proposed procedures for semantic service discovery.



(Figure 4) Detailed Procedures for Semantic Service Discovery

Finally, highly relevant, functionally feasible, and already guaranteed service snippets (i.e., reusable services which consist of more than one atomic service) are delivered to users. It should be noted that the user's experiences on the creation of service are considered directly and indirectly for semantic service discovery.

6. Use Cases

The following two use cases show expected user benefits in sharing personal service case bases in collaborative user-driven service creation environments.

1) Use Case I: Home Management using Smart TV

Let's suppose that we already have elementary APIs required to build home services. Examples are temperature sensor, heating control, alarm clock, light sensor, alarm control, blind control, indoor location, play control, TV control, etc.

Let's say a user's friend shared a service creation experience which enables home management using Smart TV. Its key function is controlling the activation (or deactivation) of home appliances for energy efficiency. However, currently it doesn't have any visualization user interface and home environment setting is not fit for the user's home management system.

In this situation, the user first adjusts number of rooms and key electronic devices to reflect his/her home environment. The user then may attempt semantic service discovery and add a service snippet to the shared one which can represent monthly (or annual) statistics of the periodically stored power consumption information. Finally, the user can add several control rules for home management. For example, there are control rule for alarming users for excessive energy use and direct control of devices according to particular situations

2) Use Case II: Dynamic Service Composition in Mobile Environments

In general, ordinary users' intents are not determined by prior history (or pre-configured user preferences). The users may want to meet serendipitous services occasionally. Although there are prototypical services that are expected in casual situations (e.g., touring, walking, driving, dining, working in the office, etc.), their satisfaction quality on the suggested services will be cranked up a notch if a minimal control is given to the users.

Mr. Kim is a university student who lives in Seoul, Korea. He would like to have dessert after dinner. However, he has no idea what options he has in which restaurants are good. Let's suppose that his girlfriend recommended him "DessertPick.com" which is highly rated by dessert lovers. He then obtained a list of relevant service snippets "DessertPick.com" is used after semantic service discovery. The most preferable service snippet of them was the one with navigator because he wanted to know how to get to the restaurant.

3) Discussion

In our proposed approaches, we assume that users interact indirectly by sharing their own or slightly modified service creation experiences. If services found as the results of

semantic service discovery do not fit to his/her current situation, the user can revise the most promising one through user-friendly service creation IDE. Users can participate more actively by borrowing others' service creation experiences, and modifying them in a personalized manner. The activity will expedite the sharing of service creation experience, and finally contribute to the ecological proliferation of user-driven service creation. Of course, SCEs that is presented to end consumers should provide ease of use, security, and stunning interactivity

However, for the maturation of seamless composition (or mashup) among existing web services, some problematic issues should be resolved. For the collaboration of different types of services (e.g., web API and telecom API), input/output data require moderate data type conversion and data mediation. When connecting SOAP-based service and REST-based service, a kind of adaptation module must deal with associated data conversion and quality problems. Aside from those data problems, compression, right management, delivery, and appropriate quality of service mechanisms should also be considered in networking domain

Meanwhile many users say that there is not enough number of reusable services that can be adopted in the context of various daily lives. This is the role of every share holder (service provider, web service portal, network operator, end-users, etc.) and their active collaboration or competition to create new business opportunity may alleviate this symptom. Its starting point is build and release 'LEGO block' like web services. Currently available web service is not straightforward. They mostly require a certain level of programming skills (e.g., Java, Python, JavaScript, etc.) and knowledge in data manipulation (e.g., regular expression, string tokenization, etc.).

7. Conclusion

In consumer networking environments for entertainment, home automation, and e-Health, users are more open to personalized service creation with the help of intuitive service creation environments. If the service creation experiences are shared efficiently, user will be more benefits through directly/indirect collaboration.

For the support of efficient sharing the experiences, in this article, we propose a novel data management system for user-driven service creation which comprises aggregation of service creation cases, data mining using service creation cases, effective utilization of the cases for semantic service discovery. To realize user-driven service computing, ecological evolution in the SCEs and efficient data management from creation to utilization are highly demanding.

References

[1] J. Sienel et al., "OPUCE: A Telco-driven Service Mash-up Approach," Bell Labs Technical Journal, vol. 14, no. 1, 2009, pp. 203-218.

[2] Y.C. Jung et al., "Employing Collective Intelligence for User Driven Service Creation," IEEE Communications Magazine, vol. 49, issue 1, Jan. 2011, pp. 76-83.

[3] C. Schroth and T. Janner, "Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services," IT Professional, vol. 9, no. 3, May-June 2007, pp. 36-41.

[4] J. Wong and J. I. Hong, "Making mashups with mar mite: towards end-user programming for the web," Proc. of SIGCHI'07, pp. 1435-44.

[5] N. Laga et al., "Widgets and Composition Mechanism for Service Creation by Ordinary Users," IEEE

Communications Magazine, vol. 50, issue 3, Mar. 2012, pp. 52-60.

[6] Y. -M. Park et al., "EasyApp: Goal-Driven Service Flow Generator with Semantic Web Service Technologies," Proc. of ESWC'11, pp. 446-50.

[7] Y. C. Jung et al., "Automatic Construction of a Large-Scale Situation Ontology by Mining How-to Instructions from the Web," Journal of Web Semantics, vol. 8, no. 2, 2010, pp. 110 - 24.

[8] Y. Hao et al., "Efficient IR-Style Search over Web Services," Proc. of CAiSE'09, LNCS 5565, pp. 305-18.

[9] N. Laga et al., "Business Process Personalization through Web Widgets," Proc. of IEEE Int'l. Conf. Web Services, 2010, pp. 551-58.

정 유 철



2005년 :한국과학기술원 디지털 미디어프로그램 (공학석사)

2011년 :한국과학기술원 전산학과 (공학박사)

2009년~2013년: 한국전자통신연구원(ETRI) 선임연구원

2013년~현재: 한국과학기술정보연구원(KISTI) 선임연구원

2017년~현재: 과학기술연합대학원대학교 과학기술정보과학과 부교수

관심분야: 정보검색, 자연어처리, 빅데이터, 딥러닝, 기계학습 등



김진영

2009년 : 서강대학교 대학원 (공학 석사)
2011년 : 한국과학기술원 전산학과 (박사과정수료)

2015년~현재: 한국과학기술정보연구원(KISTI) 연구원
관심분야: 데이터베이스 시스템, 그래프 데이터베이스, 정보 검색(IR), 빅데이터, 개체식별기술, 접근제어 등



서동준

2007년 : 한국과학기술원 디지털 미디어프로그램 (공학석사)
2014년 : 한국과학기술원 건설 및 환경공학과 건설IT융합프로그램 (공학박사)

2007년~2010년: HUMAX 소프트웨어 연구원
2014년~2015년: KAIST IT 융합연구소 연구조교수
2015년~현재: 한국과학기술정보연구원(KISTI) 선임 연구원
2017년~현재: 과학기술연합대학원대학교(UST) 과학기술정보공학과 부교수
관심분야: 빅데이터 분석, 딥러닝, 기계학습, 건설 ICT 융합 등



이혜진

2002년 : 숙명여자대학교 문헌정보학과 (정보학 석사)
2013년 : 숙명여자대학교 문헌정보학과 (정보학 박사 수료)

2002년~현재: 한국과학기술정보연구원(KISTI) 선임 연구원
관심분야: 정보검색, 네트워크분석, 계량정보 등

김광영



2001년 : 부산대학교 대학원 (공학 석사) -한글형태소분석기
2011년 : 충남대 대학원 (문헌정보 박사-개인화검색시스템)

2001년~현재: 한국과학기술정보연구원
관심분야: 정보검색(IR), 딥러닝기반 개체명인식기, 개인화 검색시스템, PLOT기반 식별기술