The incoming trends of end-user driven service creation

Zhenzhen Zhao, Nassim Laga, Noel Crespi

The Incoming Trends of End-user driven Service Creation

Zhenzhen Zhao[1], Nassim Laga[2], Noel Crespi[1]
[1]Institut TELECOM, TELECOM SudParis, Rue Charles Fourier. 9,
91000 Evry, France
{zhenzhen.zhao, noel.crespi}@it-sudparis.eu
[2] Orange Labs, France Telecom R&D, Rue des Coutures. 42,
14000 Caen, France
nassim.laga@orange-ftgroup.com

Abstract. Service creation is the next big end-user driven hype in the technology arena. As the first Web 2.0 services are already enabling tech-inclined users to build their own composite services, this trend will lead to a huge set of new features and services will be provided by all stakeholders. This paper illustrates the incoming trends of the end-user driven service creation with extensive new features, and provides five main research topics from both technical and business perspective. The underlying key technologies, as well as the current platforms/tools are also detailed.

Keywords: Web 2.0, Service creation, Semantic, Context-aware, Community.

1 Introduction

Web 2.0 puts people into the center of the vision. Communication, collaboration, collective intelligence, and social interaction between people are what Web 2.0 is about much more than the technology pieces [1]. The past few years have witnessed end-users' increasing involvement in the content creation process of modern web applications (such as Wikipedia, Blog and YouTube), which is of great benefit from collective user-generated content [2]. Currently, a growing user community is also trying to profit from existing services to generate its own applications, which triggers the trends of the end-user driven service creation.

Towards this new trend, generally, three questions are raised.

- Why should end-users create services?
- What are the ongoing works on end-user driven service creation?
- What are the challenges of enabling end-users to do future service creation?

Like different people have different personalities, the best way to achieve user personalization is giving way to end-users selves to develop services satisfying their specific needs. Telecom operators and service providers offer an increasing number of services to their customers, but it's impossible to foresee all the conditions that a customer can experience in his daily life. From the business perspective, the main idea behind the end-user service creation is to shorten the analysis/development of the marketing chain, harness the collective intelligence and promote innovation in

software development. Of course the services that end-users create won't be as sophisticated as professional services, but it will give end-users the possibility to differentiate themselves.

The service creation process is based on the concept of service composition, which allows creating/assembling a new service from existing services or a set of basic piece of services (called service enablers in Open Mobile Alliance (OMA) [3]). The guideline for designing and developing a service creation platform is "Do half work. Let users do the other half." Since the services are the results of a composition of a set of pre-defined services, the "half work" should be done is to provide the intelligent sub-services that allow for the creation and orchestration of enhanced services in a personal, dynamic and multi-domain environment.

Designing such a platform is not easy due to the complexity of current service oriented architecture (SOA) based technologies [4]. Currently programmers create easily Mashups [5], which is a customized combination of multiple functionalities of heterogeneous web sources or web services APIs. They do easily using either SOA technologies (such as REST [6] and WSDL [7]/SOAP [8]), or invoking and reading RSS feeds, or finally by invoking and extracting data from web sites through screen scraping [9]. These methods are accessible for each novice in the web development, but they are not understandable by non-developer people. Therefore, new mashup creation tools have emerged such as YahooPipes [10], MARMITE [11] and Microsoft Popfly [12] which are more designed for advanced users (tech-inclined users), but they remain not intuitive enough as ordinary users still find them difficult to use.

To reach an effective end-user service creation we have to go through several challenges. The first one consists in how to provide the intuitive and semantic tools to enable ordinary users to do creation. Following the design philosophy to be easy-to-use and to achieve a perceived usefulness, current research is making great efforts on semantic based mashups and natural language (NL) processing methods [13]. Semantic based mashups aim to enable automatic service composition, automatic service advertising (according to the current activity and context of the user), and the enhanced service discovery. NL processing methods, in conjunction with the semantic technologies, aim to provide tools that generate services dynamically according to a user NL request.

The second challenge concerns about how to construct a coherent service creation environment (SCE). It requires a certain level of harmonization between the increasing Internet/Telecom/3rd party services. It also needs to be investigated, what's needed to support and help the end-users for an easy and intuitive way to create services not only within the internet, but for all environments, they may use. This trend will lead to a huge set of new features, like context-aware service creation and social service co-creation. Effective incentives like value-added service with revenue sharing could also be used to encourage end-users to do creation. And concerning that, the trust and privacy issues should be taken into account.

In the remainder of this paper we present five future research domains of the end-user driven service creation: semantic based service creation, context-aware service creation, service creation with trust and privacy, social service co-creation and finally, service creation with revenue sharing. Fig. 1 shows the relationships between these topics. The remainder of the paper is organized by these five topics, respectively.

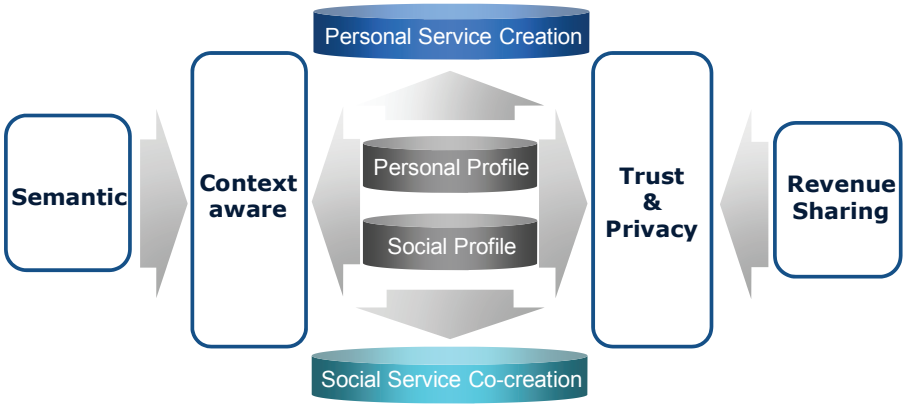

Fig. 1. Future trends of end-user driven service creation.

2 Semantic based Service Creation

Semantics allows end-user to do goal-driven service creation in a technology-agnostic way. Separating technology of a service from its meaning could improve the process of assembling new services. High performance SCE has to provide an extensible and interoperable framework able to integrate semantic field such as semantic service discovery, assembling, and publishing.

From end-users perspective, service creation process must be intuitive and self-explaining: no software development skills are required to develop a new service. End-users cannot understand current service technologies such as WSDL [7] and SOAP [8]. We witness an exponentially growing number of services on the web, which makes the end-user confused about which service to use in his SCE [10-12]. These factors make the intuitiveness almost a key for the success of service creation platform. One proposed solution is to develop a semantic service creation assistant: e.g. whenever the user picks an existing service, the system should be able to suggest a set of syntactically or semantically related services that can be connected to the existing service (Inter-service communication); or user can locate a specific service based on a description of all (or part of) its functionality with a near natural language request (NL composition [13]).

All the above processes require semantic matching, which further requires the semantic annotation of each service. Semantic annotations of the services are semantic tags/vocabularies that are integrated into service functionality. These annotations should define the interfaces of the service and the generated data. However, to enable semantic matching, the service creation platforms should either define a shared semantic vocabulary which will be used by all service providers, or to understand each semantic vocabulary of each service provider and makes translation between each other. The first solution is not dynamic as it forces each service provider to follow the platform specific vocabulary; the second one is not feasible especially for the current target where end-users become parts of service providers. In

addition, both approaches suppose the creation of a vocabulary which covers all concepts of all potential services, which is not realistic for a wide range of services.

The second challenge of semantic composition process resides in how to automatically choose (possibly discover) the specific services from the large amount of tagged services. Regarding to this, the platform should be designed taking into account the user preferences and context. Based on the contextual information, business rules or user profile, the platform should intelligently select the appropriate services for a composite service. This could be cross investigated with the context-aware issue.

In the following we will illustrate the features and future trends of NL composition, which was partly investigated by SPICE project [13]. In order to build a customized service directly from a user request, the faced challenges are:

- How to transform the user request: from natural language to formal request?
- What are the required services that respond to that request? (Service discovery)
- What is the execution sequence of these services that responds to the user request?
- And finally, does the created composite service match the user request?

By introducing Composition Factory component, SPICE composition environment summarizes perfectly the listed challenges.

Composition Factory receives a user request as an input in a formal language in which end-users can easily retrieve the requested services. This component is in charge of which composite service should be provided. First, it retrieves request-related services from the service repository and their non-functional properties from the non-functional properties component. Non-functional properties include information like efficiency, reliability and security that do not define the functions of the system but still remain important parameters in service usage. Using the user goals extracted from the request, the component builds the causal link matrix (CLM) [14] or the extended CLM (CLM+ [15]). CLM is a matrix that represents all the matching possibilities between inputs and outputs of the services, and the extended one takes into account the non-functional properties. The matching between two parameters is based on their logical relations and is statically quantified according to an ontology database, e.g. web ontology language (OWL) [16]. The quantification of semantic matching between parameters enables the quantification of the whole composite service quality. Therefore, it allows end-users and developers to select the best composite service between them.

NL composition is definitely the simplest and most intuitive way to create composite services for the end-user as it is just required to enter a NL request. However, since the web services should follow the same semantic dictionary, the semantic reasoning is a heavy task. Moreover, the created service should be validated by the end-user due to the possible mistakes of NL processing. Therefore, the NL composition should combine different technologies and skills like artificial intelligence and self-learning. The further refinement could be done in the following aspects:

User Friendly GUI Design: What graphical user interface (GUI) should be designed to help end-users to formulate their requests?

Link with Service Creation Tool: The platform should be capable of initializing a service composition definition within the service creation tool. Moreover, the run time

service composition should automatically create the user interface from the description of a requested service.

Support of Telecom Services: Telecom services such as messaging, and localization need to be integrated in the NL composition platform, so that the end-user can formulate services using these functionalities such as "Send information to me by SMS according to my current position".

Support of Multi-Language (Linguistic): The platform should support of multiple languages. Besides, adding voice can be an alternative to the NL request writing.

Support of Self-Learning: Disambiguation through recommendations and self-learning should be investigated in the NL composition platform to enhance robustness.

3 Context-aware Service Creation

Time of the old-fashioned "same static content for all users" paradigm is gone. As services are becoming more intelligent and advanced, end-users further expect the services to be discovered, created and delivered dependent on some basic information like who they are, where they are, what they have done, what they are doing and when they are doing it. This falls into the initial concern of the context-aware.

The question of how to define the context varies from different research works [17][18]. In this paper, we define context as the changing state of the user, its environment, and the interaction between them. Examples are time, place (location), user's terminal, belonging network, history of interaction and current presence. Besides, we separate the static profile from the definition of the context, examples are user identity/role and user preference.

The main features of the context-aware service creation are:

Services Selection for Composition: Intelligently choose the appropriate services adapted to the user identity/preferences/location/previous interaction.

Adaptive Interface Creation: Adapt the user interface to their current device (PC, mobile phone, PDA, TV…), browser and usage pattern.

Adaptive Notifications: Notify end-users via different channels depending on users' presence status and time preference.

Automatic Recommendations: Take into account the dynamic context information as well as the static profile of the end-user. Trace end-users' behaviors and help them to organize and filter information in order to provide personalized service/product.

Context awareness starts from capturing the users' dynamic context information. This requires the use of heterogeneous sensors distributed throughout the entire space. However, due to the fact that sensor nodes are not always available, the data aggregation mapping can change dynamically over time. Concerning that, a flexible and resilient architecture for sensor data aggregation in a heterogeneous (and dynamic) environment should be investigated. Thereafter, the context information should be interpreted and thus extract the decision to adapt or generate a service for the current situation of the user. This requires a common semantic, and decision rules between the different entities (sensors, decision maker, and the services).

The relationships and features of the semantic and context-aware service creation are summarized in Fig. 2.

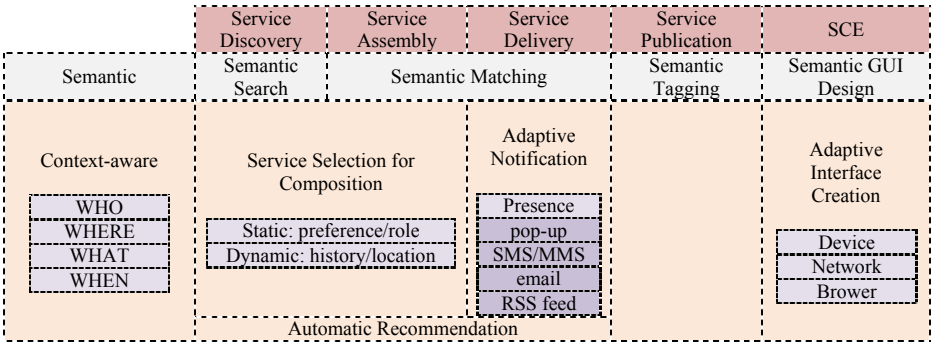

Fig. 2. Features of semantic and context-aware service creation

4 Service Creation with Trust and Privacy

Future service architecture has to proof their trust-worthiness and security awareness. Since this trend is more openness, the future services are required to be created and personalized in a privacy-sensitive manner.

Privacy protection is a cross-cutting issue since the related functionalities appear everywhere (e.g. personal data processing, shared content delivery, financial transactions). The challenges mainly concern the ability for end-users to control, maintain and implement privacy policies to protect their sensitive data in a non-intrusive manner. Therefore, an easy-to-use and intuitive user interface for manipulating their privacy policies should be considered and investigated.

Here we address the trust & privacy management issue from the following aspects:

Identity Management

The identity management is responsible for the management of the user and service attributes (logical and physical) related to the identity.

Service Identification and Collaboration: Any client (portal, service application, call centre, terminal, user agent, web service) should be identified and authenticated. This is essential for service collaboration security.

User Authentication: One identity key should be defined as the end-user's ultimate control tool over their identities. The identity key should support single sign-on (SSO) to all the services and SSO may rely on the Telco identity.

Profile Management

The profile management implies the user profile to be stored in a common representation format, and to be shared in a secure manner.

Personal Data Privacy Control: Allow end-users to have full control of their personal data. Their private data will be protected and not revealed to 3rd parties (only the virtual profile will be exposed to the public).

Content Management

Personal Content Privacy Control: Allow end-users to have the full control (includes removal) of their personal contents (media files, blog items, etc).

Shared Content Access Control: Verify the identity for accessing the shared content.

Privacy Preferences of Shared Content: Allow end-users to set different privacy preferences with respect to sharing their content.

Session Management

Session Access Control: Apply proper authorization policy (such as access, modify) to the identity who requests the access to the ongoing sessions.

Session Transfer Control: Allow ongoing sessions interacting/exchanging between different authorized devices.

Advertisement Management

Search and Receive Advert: End-users can search and receive relevant advert. The advert would take into account privacy policy settings specified in the user profile.

Retrieve and Send Advert: User profile helps advertiser to target the relevant end-users. The advert is send to end-users based on their privacy and profile settings.

Service Publication Management

Service publishing control: Allows end-users to publish the created service with a set of sharing and business policies that keep the service private for their own. Users are also allowed to change the sharing policies to make the service publicly available afterwards.

5 Social Service Co-creation

The above sections mainly discuss the "Do-it-yourself" personal service creation. In this section, we will focus on "Do-it-together" social service co-creation.

One main feature of Web 2.0 is collective intelligence. The customer relations will be the most important factors in the future networks for their survival and growth. Therefore, giving end-users the opportunity to create services together is the essential step to harness the social interaction and strengthen their relationships. The features and challenges of the future social community are:

The future social community should harmonize service creation approaches. Looking into the different vendor/project specific approaches it becomes clear, that no common approaches, even not the same kind of definitions or enabling frameworks are used. Therefore interoperability or even cross-technology service creation is not possible. In order to make service creation tools more widely useable though social communities, a certain level of harmonization between different tools needs to be achieved. This can start by merging the same kind of enabling frameworks (like OMA enablers [3] and Parlay (X) API [19]). This kind of integration can lead up to the reuse of common semantic definitions.

The future social community should provide different levels of service creation tools. Community consists of people with various levels of technical background. The service creation tools should therefore be able to operate on different technical levels. The technical minded people will be a valuable resource in developing solutions

beneficial for all members of the community. Proper tools for these advanced users will be more demanding, and allowing for more complex functionality.

The future social community should be expanding to social physical environments. Physical environment like hotels, clinics, sports bar and travel agency can be easily adapted to the physical desires of people in everyday life. On the other hand, web based social communities are well satisfied people through expressing and sharing their emotions and thoughts. Since the current social networks are proved to be a big success of strengthen user relationships, this relationship should be more exploited to meet users' necessary requirements by linking the physical environment to the social networks.

Create a Community – Contact and Social Profile Management

Our definition of community is to enable end-users to easily gather a set of group members based on their shared attributes (e.g. common interests, families, co-workers, buddies, etc). The members of community could be from Telco as well as Web world.

Contact List Control: Allow end-users to be able to manage several contact lists, e.g. create a community by choosing their friends and family members from different contact lists.

Identity Lookup: Search end-users based on their personal attributes, e.g. look up the user community ID based on their phone number.

Unified Social Profile: Provide a unified social profile shared by different networks. Moreover, end-users should be allowed to set different privacy levels to different data of his profile.

Profile Synchronization: Allow end-users to define and update their profiles in a timely manner. When a user modifies his personal profiles, the registered services and related contacts should be notified and changed accordingly.

Interaction with the Community

Social-aware Service Creation: Allow end-users to search and retrieve relevant information that takes into consideration the attributes of all community members (such as interest, presence and location). This will contribute in a created service that best suited to all the members of the community.

Autonomic Recommendation of Social Communities: Select social community for the end-users according to their static profile and dynamic context.

Interaction with Existing Social Networks: Allow end-users to interact and share data with contacts belong to the well-known social networking sites.

An example of the interaction with the social community is though intelligent calendar widgets. Since our daily lives are organized by calendar events, it's very beneficial if the participants of a calendar event can be handled as communities. Personal content could be linkable to the calendar events. Through this linking, they automatically get a context associated. All possible community-specific operations such as giving access rights to content, distributing notifications on new content will be handled automatically.

Fig. 3 further illustrates the future features and relationships between the four research topics discussed above.

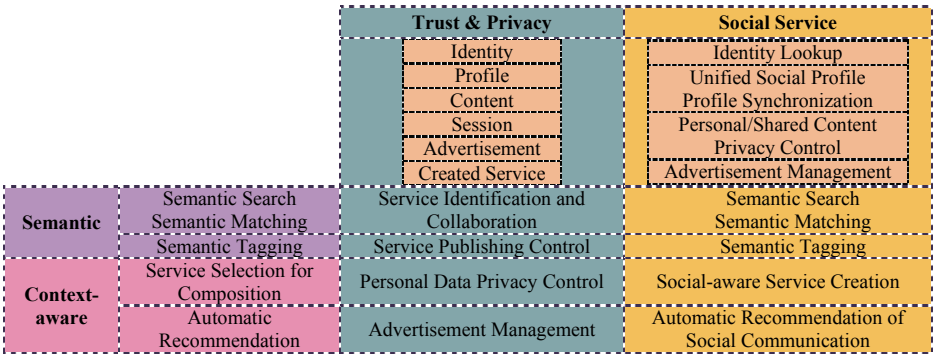

Fig. 3. Features and relationships between service creation topics

6 Service Creation with Revenue Sharing

The fifth topic is much more related to the business concern. Indeed, enabling end-users to create their own services promotes software innovation and reduces the time to market of new services. It also raises additional concern such as business models. The open business related issues are essentially focused on charging (billing), revenue sharing and QoS guarantee.

Generating revenue for the participants in a fair share could be useful to encourage end-users to do creation. However, in such multi-vendor supply network, revenue sharing model and reseller model should be defined to implement a fair division of investments, business risks, costs and revenue sharing among the different actors. Apart from this, the charging mechanisms will be revolutionary modified since the end-user (becomes one of the stakeholders) profit and risk should also be dealt with. Note that the advertisement revenue is another source of revenue, so that end-users may share the revenue through giving an opportunity of inserting an advert.

More detailed questions related to revenue sharing and charging are:

- What is the relation between the generated service creator and the composed services providers?
- How to bill end-users using a user generated service? Should service providers calculate the price or the service creator define the price by himself? In the second case, who is going to pay for the usage of the basic existing services?
- Are the used basic services billed at the creation time or at the execution time?
- How to bill a user generated service especially in the case where the used basic services have different payment method (prepaid, postpaid)?

The second raised question is how to guarantee and maintain the QoS of the user generated services. Concerning this issue, Service level agreements (SLAs) should be established between end-users, service providers and 3rd parties. Moreover the QoS features should be utilized to ensure the SLA as agreed. The exploitation of the QoS and SLA in characterization of the services will be investigated to ensure the fulfillment of the agreements in a service creation environment. In addition, the trust and privacy issues should be taken into account.

7 Conclusion

Web 2.0 is envisioned as an open garden for services. Enable end-users to do service creation can reduce the time to market of new services, harness the collective intelligence and promote innovation in software development. In this paper, we have presented an overall view of future end-user driven service creation, from both technical and business perspectives. Future trends with extensive new features have been illustrated as well as the current platforms/tools. It has been shown that the future service creation tools must be intuitive and self-explaining, and the services should be created in a context-aware way. It also becomes clear that the "Do it together" social service co-creation with revenue sharing is the next step that will happen. Moreover, in order to achieve a pervasive usefulness, a certain level of harmonization among different service creation approaches needs to be achieved.

References

1. Tim O'Reilly. What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software. (2005) http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
2. D. Braga, S. Ceri, D. Martinenghi, etal. Mashing Up Search Services. IEEE Computer Society. (2008)
3. OMA, http://www.openmobilealliance.org/
4. E. Newcomer. Understanding Web Services: XML, WSDL, SOAP, and UDDI Addison, Wesley, Boston. (2002)
5. Jin, Y. and Boualem, B. Understanding Mashup Development. IEEE Computer Society. (2008)
6. Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. Thesis dissertation. (2000)
7. W3C, http://www.w3.org/TR/wsdl/
8. W3C, http://www.w3.org/TR/soap/
9. Shu-Wai Chow. PHP Web 2.0 Mashup Projects. PACKT Publishing. (2007)
10. Yahoo Pipes, http://pipes.yahoo.com/pipes/
11. J. Wong, J and I. Hong. Making Mashups with Marmite: Towards end-user programming for the web. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York: NY, pp 1435-1444.
12. Microsoft Popfly, http://www.popfly.com
13. SPICE project: http://www.ist-spice.org/
14. F. Lécué, A. Léger. Semantic Web Service Composition Based on a Closed World Assumption" Web Services. ECOWS '06. 4th European Conference, pp.233-242. (2006)
15. F. Lécué, E. Silva, L.F. Pires. A Framework for Dynamic Web Services Composition", Halle (Saale), Germany, WEWS07. (2007)
16. OWL, http://www.w3.org/TR/owl-features/
17. H. Lieberman and T. Selker. Out of context: computer systems that adapt to, and learn from, context. IBM Systems Journal. (2000)
18. Anind K. Dey. Understanding and Using Context. Personal and Ubiquitous Computing. (2001)
19. Parlay, http://www.parlay.org/en/specifications/