Towards a methodology for prosumer service provision in ambient intelligence scenarios

Ramón Alcarria[1]*, Diego Martín[2], and Tomás Robles[2]
[1] Dep. de Ingeniería Topográfica y Cartografía
[2] Dep. de Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid, Spain
{ramon.alcarria,diego.martin,tomas.robles}@upm.es

Abstract

The difficulty of finding flexible tools for creating prosumer services is translated into a lack of tools that take into account the prosumer role and also in the lack of information on service development methodologies considering cooperation between the roles of consumers, prosumers, software developers and domain experts. Service creation for ambient intelligence scenarios adds complexity to the service lifecycle. This paper proposes a methodology for the creation of a prosumer platform for ambient intelligence scenarios. We propose to treat these services as knowledge objects to use knowledge management techniques in the interaction between different roles (users, service consumers, prosumers, advanced prosumers, platform developers and domain experts), and the development of a service provisioning platform that allows the management of the service lifecycle through defined processes.

Keywords: prosumer, mobile service provision, SPEM, prosumerization, end-user development

1 Introduction

End-user development and user participation has established itself as a new research discipline that joins the human factors of programming with the users' requirements and the functionalities of a end-user platform. From this new discipline a new types of users are appearing such as the prosumer [13] (a portmanteau from contracting professional with the word consumer) a well-known role in end-user development area; but from our researches another roles are also emerging. In this paper we present these new roles and its importance in the process of developing prosumer platforms that allow the coexistences of these roles and exploiting the benefits of the end-user participation. The Future Internet Assembly [1] mentioned this especial users, highlighting the need of integrating them in the process of platform development and services provisioning.

We present in our research work the service prosumerization as the process leveraging end-users to build services that solves basic problems in a IoT domain with the specific functionalities provided by the platform; allowing the users to create, share and extend new functionalities. For this reason, the specific functionality provided by the services built by prosumers depends on the global functionalities, operations, actions and objects provided by the prosumer services provisioning platform.

From our researches, we devised new differents kinds of roles that participate in the services prosumerization process that having special requirements [10] not solved by current development methodologies. In this paper we present these new roles and its importance in the prosumer platform development; a lifecycle for the prosumer services that take advantage of the global collaboration and knowledge

IT CoNvergence PRActice (INPRA), volume: 3, number: 1 (March), pp. 3-16

*Corresponding author: Departamento de Ingeniería Topográfica y Cartografía, Universidad Politécnica de Madrid, Avenida del Mediterráneo km 7,0, 28031 Madrid, Spain, Tel: +34-91-336-6487 sharing among prosumers; and an approach of a methodological development for prosumer platforms. Apart from our previous works [13] [5], there are studies that explicitly take into account the new prosumer roles but only as customer involvement in the business process [3]. Nor is there much information about the prosumerization of an existing platform and be benefited of the end-user participation and development.

This paper extends our previous work [12] that showed an a 4-leaf clover shape methodological approach for the prosumer services development. We presented six roles that participated in the definition and implementation of a prosumer platform: users, service consumers, prosumers, advanced prosumers, developers and domain experts. Taking into account these roles, we considered services as knowledge objects [14] [15] to present a service lifecycle where the possible states of a service are defined. In this new work we define from the formal point of view the processes involved in the life cycle of a prosumer service and establish through these processes four use cases that show the relationships between the different roles. These four use cases are: creation from scratch, creation from template, service execution and service exploitation. We use SPEM for this formal definition language, a modeling language commonly used in software engineering for defining processes and methodologies.

The rest of the paper is as follows: Section 2 describes the prosumer model differentiating from traditional service provision models. Section 3 presents the defined roles in prosumer environments, and Section 4 the prosumer service lifecycle and how processes and roles interact. Section 5 describes some conclusions of our work.

2 Prosumer model and co-creation

Prosumer users are involved in uploading information to the Web and, at the same time, consuming information, creating virtual communities. The success on this communities and the revolution of the Web 2.0 suggest that the prosumer model is gaining momentum. An example of this evolution can be seen in the creation of communities oriented to a prosumer market (Ebay), others that enable the publication of multimedia content (Youtube) and others that enable the interaction to virtual objects (Second Life).

This communities often provide traditional services and a reduced number of business exploitation models. In particular, service definition, pricing, authentication, authorization, and accounting are performed exclusively by a specific service provider, in some cases an operator. The term co-creation, as an strategy to bring different parties together (a company and a group of customers) to produce a mutually valued outcome, often appears in this communities but generates many problems to be solved.

One of them is the dramatic increase of management resources required to provision a large variety of micro-services, customizable by users. New business ideas are emerging for these scenarios, such as tradable microservices, which allow to be handed from one actor to another by a transfer of ownership accompanied by some monetary exchange.

Co-creation is also applied depending on the complexity the market offering, as the product or service delivered to the customer. In high-complex market offering scenarios (engineering projects, consulting, hardware solutions, etc.) companies are about to shift from competition based on product differentiation to provide a customization solution with the help of end-users [8]. Moreover, in professional services, the customer often actively participates in the process of defining the service specification, requirements and processes, as Silvestro [23] states.

In software development scenarios prosumers are studied from the perspective of end-user software engineering. End-user software engineering considers that prosumers are different from traditional software engineers. Thus, traditional approaches would not be likely to produce successful results. One reason is that prosumers often have very different training and background than professional programmers. Even more important, prosumers also face different motivations and work constraints than professional programmers. They are not likely to know about quality control mechanisms, formal development processes, modeling diagrams, or test adequacy criteria, and are not likely to invest time learning about such things. This is because in most cases, prosumers are not striving to create the best software they can; rather, they have their "real goals" to achieve: such as accounting, teaching, managing safety, understanding financial data, etc.

Summarizing, the new prosumer model presents some features that distinguish it from other traditional service provision environments:

- Non-expert users, ease of service creation and provision of a semiautomatic model for managing prosumer services.
- Several user roles (consumers, prosumers, advanced prosumers, domain experts and platform developers) and several types of services imply different communication models (P2P, collaborative, client-server).
- Interoperability of services created by different users: enabling services created by different users to share information and to interoperate (transparent combination of multiple information sources).
- Discovery, recommendation and searching of relevant mobile services: The mobile aspect of the creation of services makes the immediacy in the search be a key factor.

The main objective of a prosumer platform is to bring to the end user the possibility to create by themselves new functionalities in ambient intelligence scenarios. In a classical development approach, methodologies consider the user as a role that will use the final product with the features described in the requirements. A user-centric platform must be designed taking into consideration the type of prosumer to be involved into service development. We propose a methodology for prosumer service provision based in our previous experiences in designing and developing prosumer platforms such as GISAI-Pharma [4], DM-SePPs [11], SASEP [17] or other user-centric frameworks like holistic frameworks [21] or goal-oriented frameworks [9].

3 Roles in prosumer platforms

In this section we present six different roles that we devised in our researches as shown in Figure 1. It is necessary to remark that is a UML use case diagram, but in this case represents the functionalities on how the users use the platform tacitly. That is, the platform is not designed to for that particular types of users, otherwise it is a way of representing the different roles, that we have observed, describing the type of involvement with the platform. The platform is designed so that there is only one type of user: the prosumer. The first two are the least involved with prosumers platforms: users and services consumers; the user performs a classic usage without unveiling the potential of the platform; consumer services, only use services created by others. The prosumers and advanced prosumers, are able to use the full potential of the platform; while the prosumers are capable to build its own services; advanced prosumers are able to give sufficient quality to their services for publishing and selling them. Domain experts provide to the platform developers the knowledge and the requirements for creating the platform, ideas for new functionalities, and describe the objects to be controlled by the platform, etc.

Users: Users are involved in the basic functionalities of the platform without the benefits of creating and using prosumer services. Users do not search the repository looking for a service that solves a specific problem because they do not understand the platform environment where they can build services to solve problems. The platform is focused in more advanced users, ie prosumers. Therefore, these users must be helped and educated to become prosumers. To speed the evolution of the users we suggest to train them

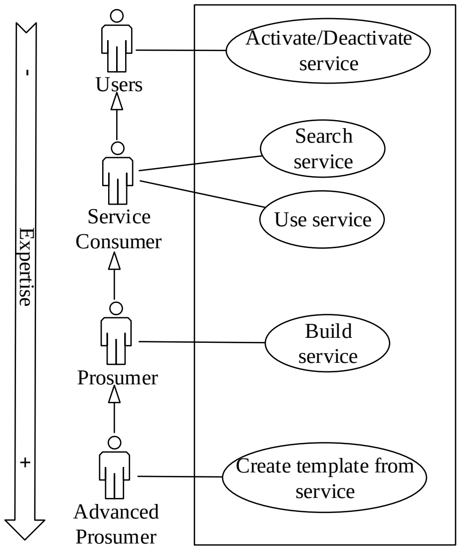


Figure 1: Prosumer development - interacting roles

through courses, workshops, practical experiences, etc. These training sessions are recommended to be led by prosumers from the same organization and supervised by developers.

Service consumers: As the name implies, they are users who consume services produced by a prosumer but they are not involved in the creation of these prosumer services. As well as users, they must be motivated by training to be able to get the maximum benefit from the platform.

Prosumers: This is the desired type of user for these kind of platforms. They have the ability to consume and produce prosumer services. Normally, they consume their own services but they can also use services produced from others. They are different from the two previous kinds of users because of their producing intentions. It is important to note that they are not the platform developers; the prosumers are not involved in the development of the platform, they build and run services on top of the platform.

Advanced prosumers: They are a special type of prosumers who have more experience or skills in using prosumer platforms. They create their services with such quality that they can be shared with other prosumers. Generally, a prosumer create a service targeting a specific problem, but sometimes, if the service is designed in a more general way, it can be used to solve a set of problems related to an scenario. In this case the prosumer service becomes a prosumer template, which can be personalized by other prosumers to create final services to solve specific problems. To make the prosumer evolve to an advanced prosumer, as always, we suggest training. Advanced prosumers can also conceive new requirements for the platform as well as add-ons, plugins, modules, operations, etc. proposing these new requirements to developers.

Domain experts: They are specialists in the domain and they know how services can be useful in the domain to solve problems and to contribute to the business model. They often propose functional requirements and new features of the platform. It is very important to take into account their proposals because they know the business in a more strategic and long-term way; they know the new features to be developed due to their executive vision of the platform.

Platform developers: they contribute to provide the platform so that it can be the interaction point of the rest of the roles. To do that we consider requirements proposed by users, prosumers, advance prosumers, and domain experts. They are expert developers with high experience in application developing and software development methodologies.

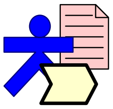
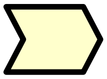
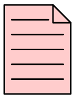
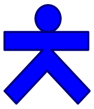
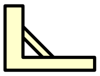
Element	Symbol	Description
Process		A process is a decomposable component and executor of the work definition that represents a relationship between tasks and roles.
Activity		An activity is a concrete definition of work that represents a general unit of work assignable to specific executors by the use of one or more roles. Defines basic units of work within a Process as well as the Process itself
Task		Describes an assignable unit of work. Every Task Definition is assigned to specific Role Definitions. A Task is associated with input and output Work Products.
Product		A work product is an item that is used, modified and generated by Tasks.
Role		A role is an element that is defined as a set of skills, competencies and responsibilities. The roles are used by the tasks to define who executes them, also to define a set of work products responsible for these roles.
Guidance		A guidance is a describable element that provides additional information related to activities and tasks. Some examples of guides can be guidelines, templates, techniques, mechanisms, etc.
Product		A work product is an item that is used, modified and generated by Tasks.

Table 1: SPEM element description

4 Prosumer service lifecycle

This section defines the lifecycle of the prosumer service, as a series of interrelated processes, defined using SPEM, particularly SPEM 2.0, which is a modeling language used in software engineering for defining processes and methodologies.

SPEM (Software Process Engineering Meta-Model) [2] is a specification of the OMG (Object Management Group) used to define software and systems development processes, as well as their components. Its main objective is to cover a wide range of development methods and processes of different styles, levels of formalism and life cycle models, among others. SPEM elements used in this section are listed in Table 1.

For the management of the lifecycle, works related to knowledge management are considered, as a prosumer service can be seen as a set of elements that are related according to a knowledge transferred to a service logic language by the creator of the service. This knowledge is distributed, used, adapted and reused in different phases of the service lifecycle. We rely on the work of Nonaka and Takeuchi [18] and the work of Rus and Lindvall [19], who consider knowledge assets should be managed according to their life cycle. As knowledge objects, prosumer services must be managed by describing **when** and **how** to apply the service and the **resulting context** of using the object. In this paper we present a service lifecycle composed of eight states (Figure 2).

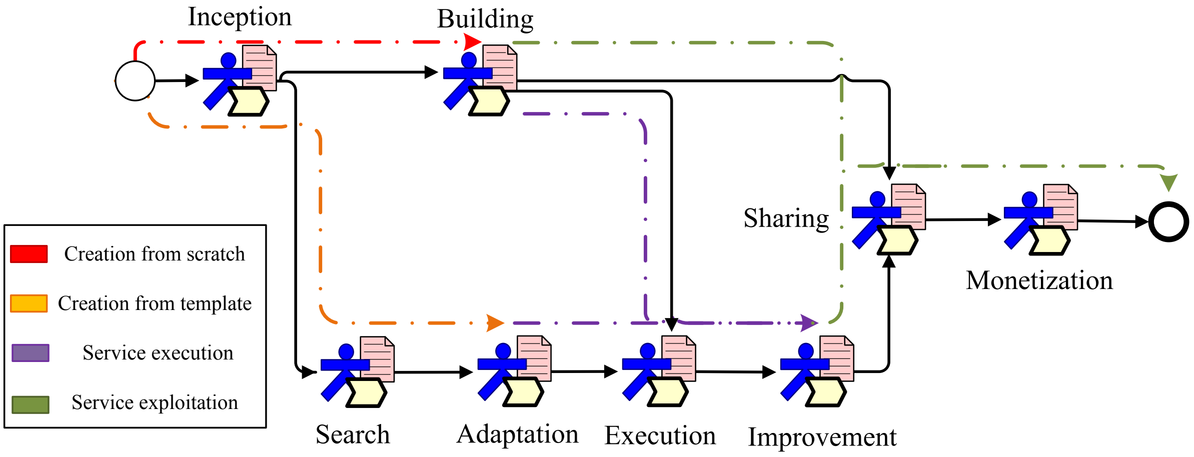


Figure 2: Prosumer service lifecycle

4.1 Process composition

Prosumer service lifecycle described in the previous section consists of eight activities that can be grouped into four processes. In this section we formalize these processes in SPEM, breaking them down into tasks, roles, products, guides, and tools. In this way we can model the interaction between roles for each one of the defined activities.

4.1.1 Creation from scratch

The creation from scratch process is decomposed in three activities according to the SPEM diagram of Figure 3: Service needing Identification, System element identification and Service Building. Service needing and System element identification belong to the same activity “Inception”.

The input products are domain experience and service creation experience, in addition to a guidance consisting on training material generated from the training stage in the method. The three defined activities are decomposed into tasks that are assigned to the Prosumer and Advanced Prosumer, and exchange work products according to this Figure.

Inception: Services are created to cover a requisite of fulfil an idea. Service functionalities are dependent on platform operations. Thus, services to be created are limited by functionalities offered by the platform. In the service inception process prosumers and domain experts determine which services can be created, so that they can proceed to the following process, which can be either “Service Building” or “Search” depending on the intention of the user to reuse an existing service (Creation from template) or to start a the building of a new service from scratch.

Service needing identification: This task starts once the prosumer has decided that the must create a service that meets his objectives. To do that, he has the knowledge taken from his experience in the application scenario. In this task the prosumer devises the service needed to solve a problem The outcome of this task is a set of creation requisites, to be used as input in the next stage.

System element identification: This task starts once the Service needing identification task finalizes, and in this task the prosumer identities the elements that offer the framework to meet their service needs.

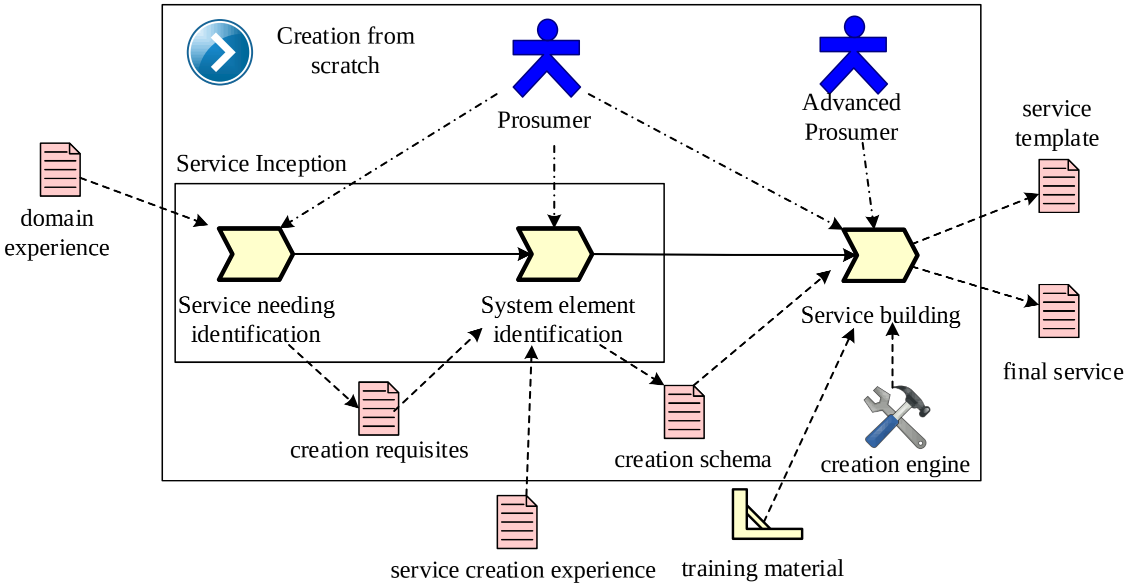


Figure 3: Creation from scratch SPEM diagram

The income of this task is the experience of the prosumer using the framework and a set of creation requisites, which are transformed to a creation schema, that is, the idea of how to compose the service taking into account the elements supported by the framework.

Service building: The service begin to exist in the building process in which prosumers create their services using some tools provided by the creation platform and by using the information devised in the inception process. They create services from scratch using the functionalities, operations and objects controlled by the platform. As an input to perform this task, the user has the work product creation schema and also the training material, modeled in SPEM as guidance, which was provided in the training stage of the framework and has been improved over the various service creation iterations.

A common issue in this step is that prosumers get stuck in some part of the creation process, due to a lack of complete knowledge of the creation environment, lack of expertise and/or expert guidance. Thus, an advanced prosumer can assist the prosumer when he/she gets stuck.

The outcome of this task, generated with the help of a tool called creation system, is a service template or a final service, depending on the creation strategy implemented in the framework. Advanced prosumers or domain experts can propose new operations, functionalities and new elements to be controlled by the platform; collecting new requirements for the platform developers.

4.1.2 Creation from template

The creation from template process is decomposed in three activities according to the SPEM diagram of Figure 4: Inception, Search and Adaptation. In this diagram we do not decompose Inception into Service needing and System element identification as they were already explained in the previous section.

The search process allows to consumer to retrieve a service or template from an external repository. This process considers only one task, the search task. The work product search criteria and the repository information are needed by this task, which may be performed by the consumer but also by a prosumer acting as a consumer. The output of this task is a service or template, delivered by a tool called search system.

Search: The search process allows to consumer to retrieve a service or template from an external

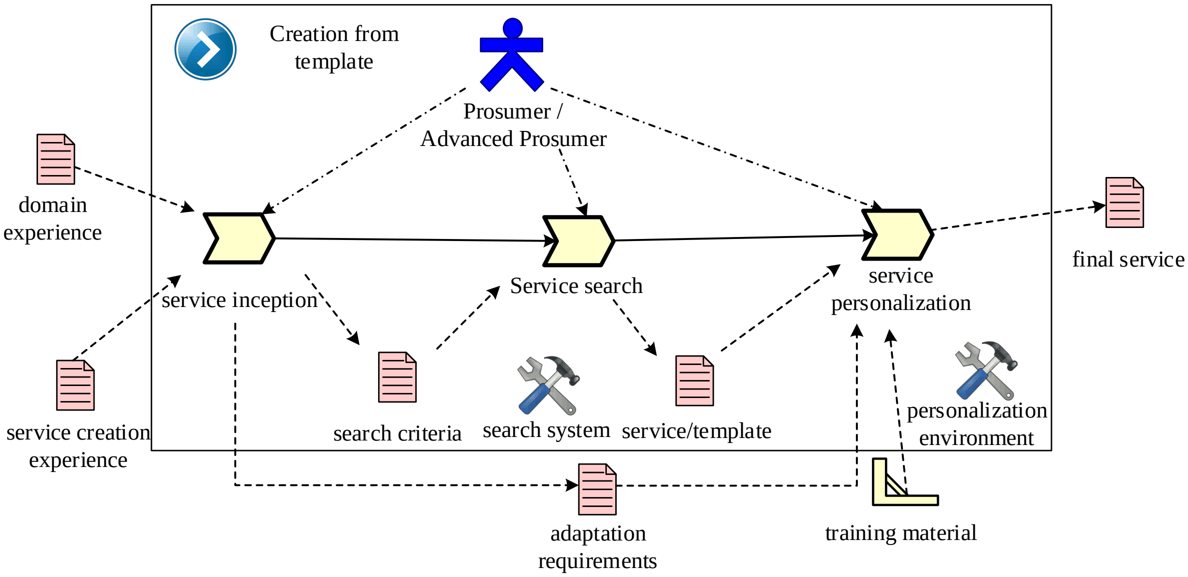


Figure 4: Creation from template SPEM diagram

repository. A service must be shared in order to be found by another prosumer. We found that it is preferable to use and customize a service already created instead of creating one from scratch because, apart from the advantage of reuse, already created services have been tested by the community, and are likely to have better quality. The platform may offer a system to find these shared services, using different query elements such as textual queries, graphical queries, queries as examples, etc.

Adaptation: The adaptation task starts once received the service template. It is difficult that a shared service fulfills all the user needs. In the adaptation process, prosumers that selected a shared service must make the necessary changes to it by using the platform. First of all, it is needed to determine what level of adaptation requires the service template and how service personalization should be to meet user goals. To do that the work products domain experience, service creation experience and the service template itself are needed. Then, the prosumer performs the personalization over the service template. As an input for this task the user has the adaptation requirements work product and also the training material, modeled as a guidance in SPEM. The training material was provided the the training stage of the framework and has been improved over the various service creation iterations. The outcome of this task is the result of the creation from template process, i.e. the final service, which can be republished. Service adaptation is different from service improvement, as adapted services are not better than the original, they just solve a different problem.

4.1.3 Service execution

The execution process (Figure 5) is decomposed into two tasks, service execution and improvement, which are assigned to the role of the user or users executing the service; in this case it may be either a prosumer a consumer, or a basic user.

Execution: Once a service is created, retrieved or customized it is ready to be executed. The platform must offer a system to control, execute and monitor the different services running on the platform. The execution process receives the work products executing participants and service and generates as a result the execution report and execution results work products. The execution report work product is generated by the user manually or semi-automatically and contains non-technical information about the service

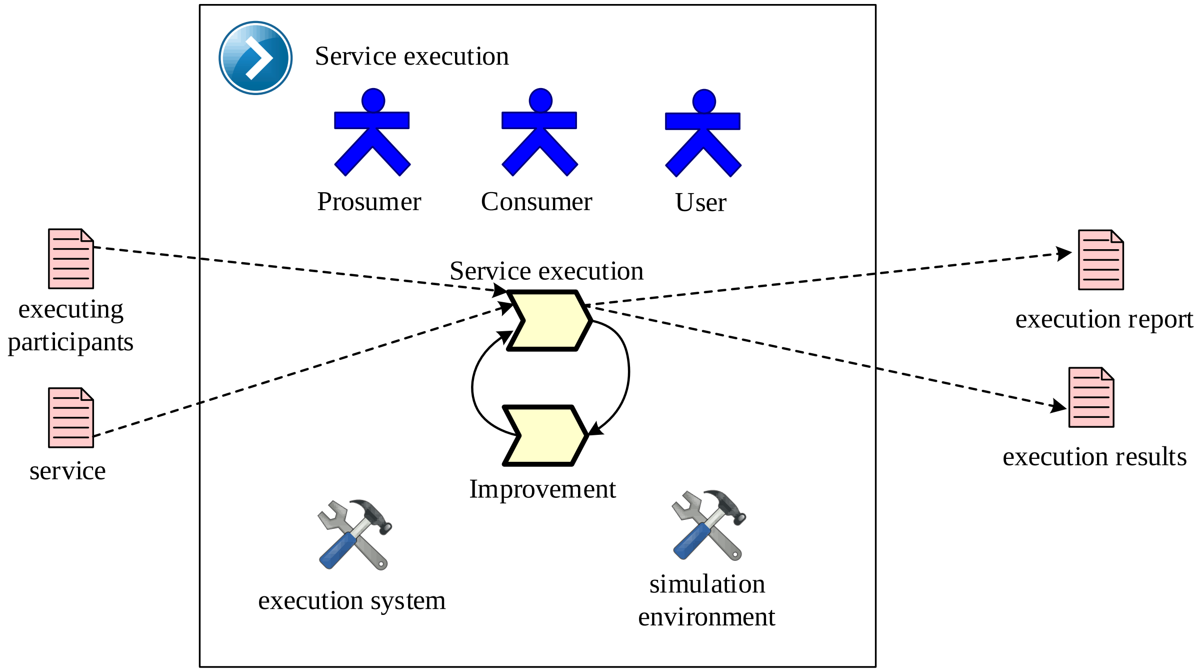


Figure 5: Service execution SPEM diagram

performance and usefulness. The execution results work product is an automatically generated report that contains information from the execution log or monitoring activities occurred during the execution of the service.

Improvement: Developed services require debugging and improvement. The improvement task is executed along with the execution task in an iterative way. In that way, as prosumer services are created by non-expert developers it is important to follow an iterative lifecycle to obtain sufficient quality. Providing a simulation environment is an specific way to test service execution while prosumers learn how to improve the quality of their services.

4.1.4 Service exploitation

The exploitation process allows us to share services that were previously created/executed and to get value from these services, the creation and execution processes themselves. For this, the exploitation process is divided into two tasks, service sharing and monetization. Figure 6 shows the process modeled in SPEM.

Sharing: At this state, the prosumer can decide to share the created services, making it available for other users. The main goal of the sharing task is that the generated/executed service will be available through a service repository so that potential users can consume it. This task requires as input the service to be published, information about its execution and information of the repository to perform the publication. In order to be shared, the service must have been used and tested, demonstrating sufficient quality. Thus, users who share services are advanced prosumers.

Monetization: Services with sufficient quality must be monetized, that is, the different roles that contributed to create and improve the service must get a benefit from sharing it. These roles are the ones who participated in the service lifecycle and contributed to the services, i.e. the prosumer/advanced prosumer, domain expert and platform developer. Service monetization is very complex and must be

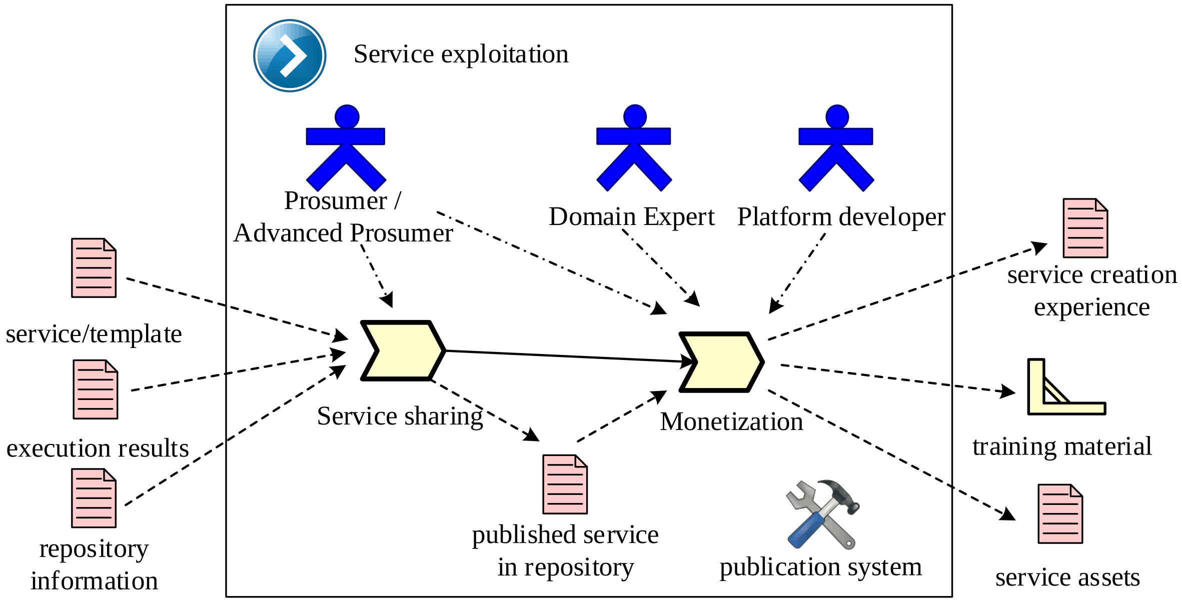


Figure 6: Service exploitation SPEM diagram

explained in the business model, defined before starting the service lifecycle.

4.2 Role interaction

Users of the described platform are mainly prosumers, advanced prosumers, platform developers and domain experts. Figure 1 shows, in addition to these roles, users and service consumers. However, these two roles are not taken into consideration because the platform is designed to allow prosumers to take advantage of service prosumerization, therefore users, and service consumers must evolve to prosumers or advanced prosumers in order to unleash their potential. This evolution will occur through training as discussed below.

Describing more specifically this type of development, a platform for services provisioning, it would be a small or medium-sized development centered on quality, user-centric, test-driven, with continuous integration. From our experience, we recommend the use of an agile methodology such as XP [7], Kanban [22], Scrum [20], Agile RUP [6], etc. In the case of developing a prosumer platform we suggest to use Scrum because we believe this technique is the most useful in this kind of developments where it is needed a high integration with the final users. Classical methodologies are not recommended for this type of platform considering that they are not taking into consideration the final user participation and their integration with software development processes. Thence, we will present each of the tasks performed by the four main roles.

4.2.1 Developers

Developers consider the requirements proposed by other roles and, using their high skills in software development, provide various version of the platform and some other extensions as add-ons.

Platform provisioning: Developers analyse the requirements proposed by other roles such as domain experts, users, managers, etc. With these requirements they design, implement and deploy the platform. They develop the platform following the phases proposed by any methodology, classical, agile, spiral, waterfall, etc. It is necessary to trace the requirements and who was the source of these requirements, because it is fundamental to assist prosumers, advanced prosumers and domain experts as it is necessary to guarantee a user-driven development.

Add-ons development: Developers not only develop the platform, they are in charge to develop new add-ons and other features and, of course, its maintenance. The requirements for these new functionalities should be provided by prosumers, and domain experts.

4.2.2 Prosumers

In a user-centric development the main actors are users, but in a prosumer platform these actors evolve from users to prosumers. They have the ability to consume and produce prosumer services.

Training: The best way to stimulate the evolution from users to prosumers is through training. With training the normal users, service consumers and prosumers can evolve into an actor who is more involved with the software development. We suggest that the training process should be led by developers and advanced prosumers as, due to their experience, they can help with the training sessions. Sessions must be efficient, pragmatic, amusing and participative among all the actors in order to take advantage of the whole platform.

Service devise: Knowing the possibilities and features of the platform, a prosumer can think out a problem he wants to solve. The first thing they should do is to search in the repository for some services that resembles what they want to create; if they don't succeed, they can create a service from scratch or based on a template. If they cannot find the right functionalities to create the service because the platform does not offer them, they should propose these functionalities to the development team in order to be studied and marked for their implementation.

Building: In this phase, prosumers use the platform and its possibilities to create services without the help of the developers. Prosumers are in charge of populating the repository with services. While it is true that developers may offer some service examples, proposed by the domain experts, as starting point for new services.

Evaluation: The best way to know if a service meets its requirements is to evaluate it measuring some indicators. These indicators reveal important information about the service, they can show if a specific services is well built, if it is used or whether the platform needs more functionalities to be developed.

4.2.3 Advanced prosumers

This special type of prosumers personalize general services to solve specific problems. They also conceive new requirements for the platform and foresee the most suitable business model for the developed applications.

Improvement: Advanced prosumers are those prosumers with sufficient experience who are able to give more quality to the services they create. They can perform various types of improvements.

- The first ones are improvements in services themselves. They can build abstract services that are able to solve more general problems; so a prosumer or service consumer can choose one of these abstract services as a template to start building a new service.
- The other improvements are related to the information improvements contained in each of the knowledge elements of the service. Advanced prosumers can change service metadata to improve service indexing and searches made by other users, thus, producing services that can be found more easily.
- Even more, advanced prosumer can use a service simulator to test the improvements and check whether the service can be published and monetized.

Sell: Advanced prosumers can think about getting money through the services they created by publishing them in service repositories, hoping they can be useful to other users who are willing to pay for a service that solves their problems.

4.2.4 Domain experts

These domain specialists propose functional and non-functional requirements and new features to the platform, which can be translated to new functionalities or new add-ons depending on whether these features are considered to be a core part of the platform or an interesting non-core item.

Platform requirements: The primary source of requirements are domain experts. They provide the guidelines for creating the platform. But they not only provide these requirements at the beginning of the project when the platform is only an idea; they also propose new requirements when the platform is working and new ideas arise about functionality.

Functionality devise: The main functionality of the platform is to manage prosumer services and execute them in an IoT and ambient intelligence scenario. These scenarios tend to change very quickly, appearing new objects to be controlled by the services and operations to be performed. Domain Experts are the best people to detect these new needs.

New Add-ons proposal: Domain Experts may provide new functionality to be implemented by the platform due to the fact that ambient intelligence scenarios are continuously changing. They can propose new add-ons to the platform, but the can also highlight unused, useless or add-ons with problem, identifying that unused functionality.

5 Conclusions

This paper presents a methodology for the creation of a prosumer platform for ambient intelligence scenarios, as a reliable solution that will be more widely used than traditional software engineering techniques as it considers service co-creation to generate value-added services. As the development and maintenance of this platform may be complex due to the variety of roles and the interaction between them we propose in the paper a new set of roles to be considered, a service lifecycle considering services as knowledge objects [16] used in these platforms and a methodological approach to develop and maintain a prosumer platform for ambient intelligence scenarios. These proposals are the result of our experiences in developing such platforms. This methodology is being validated for the development of applications to be deployed in a supermarket domain, in which prosumers with specific domain knowledge are involved as well as developers.

As future works we propose to conduct a research experimentation to prove that the use of these three elements (roles, service lifecycle and methodological approach) proposed in this paper really offers improvements over using other classical (agile or not) methodologies. We will compare this methodology to other methodologies that do not take into account the prosumer concept, co-creation, and end-user development.

Besides, we plan to incorporate some Scrum-based software development techniques, based on iterations, focusing on the implications of having this new role on sprints and project cycles.

Acknowledgments

This work has been partially supported by project CALISTA (TEC2012-32457) and also by the Autonomous Region of Madrid through programme MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER).

References

[1] Future Internet Assembly Research Roadmap v2.0. Available at the European Future Internet Portal: `http://www.future-internet.eu`, last viewed February 2015.

[2] Software & Systems Process Engineering Metamodel Specification (SPEM) Version 2.0. Release Date: April 2008. Available at: `http://www.omg.org/spec/SPEM/2.0/`.

[3] Y. Akao. *Quality function deployment: integrating customer requirements into product design*. Productivity Press, Cambridge, Mass., 1990.

[4] R. Alcarria, D. Martín, T. Robles, and A. Morales. A case study for validating a prosumer framework in drug traceability scenarios. In *Proc. of the 7th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI'13), Carrillo, Costa Rica, LNCS*, volume 8276, pages 311—-318. Springer International Publishing, December 2013.

[5] R. Alcarria, T. Robles, A. Morales, and S. González-Miranda. New service development method for prosumer environments. In *Proc. of the 6th International Conference on Digital Society, Valencia, Spain*, pages 86–91. IARIA, January-February 2012.

[6] S. W. Ambler. *The Enterprise Unified Process: Extending the Rational Unified Process*. Prentice Hall Press, 2005.

[7] K. Beck. *Extreme Programming Explained: Embrace Change*. Ed. Addison-Wesley Professional, 2001.

[8] R. Dhar, A. Menon, and B. Maach. Toward extending the compromise effect to complex buying contexts. *Journal of Marketing Research*, 41(3):258–261, August 2004.

[9] J.-I. Fernandez-Villamor, C.-A. Iglesias, and M. Garijo. A framework for goal-oriented discovery of resources in the restful architecture. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(6):796–803, June 2014.

[10] Álvaro Carrera, C. A. Iglesias, and M. Garijo. Beast methodology: An agile testing methodology for multi-agent systems based on behaviour driven development. *Information Systems Frontiers*, 16(2):169–182, April 2014.

[11] D. Martín, R. Alcarria, Álvaro Sánchez-Picot, and T. Roble. Decision making environment based on prosumer services for tracing drugs in a hospital pharmacy department. In *Proc. of the 8th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI'14), Belfast, UK, LNCS*, volume 8867, pages 324–332. Springer International Publishing, December 2014.

[12] D. Martín, R. Alcarria, Álvaro Sánchez-Picot, T. Robles, and D. S. de Rivera. A four-leaf clover shape methodology for prosumer service developments. In *Proc. of the 8th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI'14), Belfast, UK, LNCS*, volume 8867, pages 488–495. Springer International Publishing, December 2014.

[13] D. Martín, R. Alcarria, T. Robles, and A. Morales. A systematic approach for service prosumerization in iot scenarios. In *Proc. of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS'13), Taichung, Taiwan*, pages 494–499. IEEE, July 2013.

[14] D. Martín, J. G. Guzmán, J. Urbano, and A. Amescua. Modeling software development practices using reusable project patterns: a case study. *Journal of Software: Evolution and Process*, 26(3):339–349, March 2014.

[15] D. Martín, J. G. Guzmán, J. Urbano, and J. Llorens. Patterns as objects to manage knowledge in software development organizations. *Knowledge Management Research & Practice*, 10(3):252–274, May 2012.

[16] M. Merrill. Knowledge objects and mental models. In *Proc. of the 2000 International Workshop on Advanced Learning Technologies (IWALT'00), Palmerston North, New Zealand*, pages 244–246. IEEE, December 2000.

[17] A. Morales, R. Alcarria, D. Martin, and T. Robles. Enhancing evacuation plans with a situation awareness system based on end-user knowledge provision. *Sensors*, 14(6):11153–11178, June 2014.

[18] I. Nonaka and H. Takeuchi. *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995.

[19] I. Rus and M. Lindvall. Knowledge management in software engineering. *IEEE Software*, 19(3):26–38, May-June 2002.

[20] K. Schwaber. *Agile Software Development with Scrum*. Pearson Education, 2008.

[21] E. Serrano, G. Poveda, and M. Garijo. Towards a holistic framework for the evaluation of emergency plans in indoor environments. *Sensors*, 14(3):4513–4535, March 2014.

[22] S. Shingo. *A Study of the Toyota Production System from an Industrial Engineering Viewpoint*. Productivity Press, 1989.

[23] R. Silvestro. Positioning services along the volume-variety diagonal: The contingencies of service design, control and improvement. *International Journal of Operations & Production Management*, 19(4):399–421, 1999.

Author Biography

Ramón Alcarria received his M.S. and Ph.D. degrees in Telecommunication Engineering from the Technical University of Madrid in 2008 and 2013 respectively. Currently, he is an assistant professor at the E.T.S.I Topography of the Technical University of Madrid. His research interests are Service Architectures, Sensor Networks, Service Composition and Prosumer Environments.

Diego Martín received his doctoral degree in 2012, holds a B.Sc in Computer Engineering and an M.S. in Computer Science from the Department of Informatics at the Carlos III University of Madrid, Spain. His main research areas are Software Process Improvement, Knowledge Management and Reutilization and Prosumer Environments.

Tomás Robles received a M.S. and Ph.D. degrees in Telecommunication Engineering from Technical University of Madrid in 1987 and 1991 respectively. Since 1991 he is associate professor on Telematics Engineering at the E.T.S.I. Telecommunication of the Technical University of Madrid. His research interest is focused on Advanced Applications and services for Broadband networks, both wired and wireless networks.