An Architecture to Support the Design and Evaluation of Software Platforms for Partially Virtual Communities

Francisco Gutierrez[1], Sergio F. Ochoa[1], Nelson Baloian[1], Gustavo Zurita[2], and Luis Loyola[3]

[1]Computer Science Department, Universidad de Chile

Av. Blanco Encalada 2120, 3rd Floor, Santiago, Chile.

{frgutier, sochoa, nbaloian}@dcc.uchile.cl

[2]Management Control and Information Systems Department, Universidad de Chile

Diagonal Paraguay 257, 13th Floor, Santiago, Chile.

gzurita@fen.uchile.cl

[3]R&D Department, SkillUpJapan Corporation

Tokyo, Japan

loyola@skillupjapan.co.jp

Abstract. Several researchers have identified the design of collaborative systems as a particularly challenging task because they involve several services that are not easy to identify by software developers. The systems that support the activities of a partially virtual community (PVC) are not an exception. Trying to contribute to address such a design challenge, this article proposes a software architecture that can be used for both, designing new supporting systems and evaluating the existing ones. The architecture was used in three case studies as a design guideline during the development of PVC supporting platforms, and also as an instrument to evaluate three commercial systems. Although the obtained results are still preliminary, they indicate that the architecture is able to accomplish both roles.

Keywords: Social system architecture, design guideline, evaluation instrument, partially virtual communities.

1. Introduction

For the past twenty years, researchers have been studying and classifying virtual communities in several scenarios [Hunt09] [Plan04]. One particular type of these associations is what we call *Partially Virtual Communities* (PVC) [Guti12], where the community members have the opportunity to interact frequently through both a virtual and a physical space. Typically members know each other, and such contextual information allows them to appropriately interpret others' contributions. Membership in these communities is quite stable, meaning that few people join or quit them. The personal connections among members make these communities stronger and tightly linked.

Although participants in a PVC cannot easily leave the group due to their personal ties, several studies indicate they ignore the community when a number of conditions are not met. When this members' attitude becomes regular, the community typically deal to its demise [Irib09] [Mous09] [Hill09].

Every PVC evolves through time, therefore some supporting services used by a community become obsolete and also new services are required according to the community maturity level. Software designers have to identify the services currently required by the community and envision those eventually required in the near future, as a way to prepare the supporting platform for the next evolution stage in the community lifecycle. To the best of our knowledge, the literature does not report any work presenting structural designs that help address this challenge; therefore designers of this type of systems must improvise or adopt ad hoc solutions to deal with this issue. Moreover, software developers typically have to face the *iceberg effect*, that is usually present in the development of collaborative systems [Hers11]. Using the metaphor of an iceberg Herskovic et al. [Hers11] show that developers usually focus their efforts in services that have a visual representation on the user interface, and they tend to overlook critical (mandatory) requirements that do not have a visual representation. This lack of an explicit presence of in the user interface makes these services not easy to identify. Examples of them are the community governance and monitoring, and also the tracking of members' activities. If designers have no experience modeling this type of systems before, it is highly probable that several of these services will be not considered in the design of the software platform supporting the PVC. A deficient design will lead the system towards a limited implementation, which will affect negatively to the community.

In this article we propose a software architecture that helps design PVC supporting platforms and evaluating already implemented systems. Particularly, this architecture offers support when (1) designing this kind of systems, through the use of general guidelines and giving specific feedback to developers about services to be considered in the new supporting platform; and (2) evaluating already implemented systems, by comparing the typical supporting services required by a PVC and the services provided by a certain platform. The usefulness of these services has been evaluated by members of a PVC and also by developers with highly encouraging results.

Next section introduces the concept of a PVC and its main characteristics. Section 3 presents the related work. Section 4 describes the requirements that are typically involved in the support of the PVC activities. Section 5 presents the proposed software architecture and describes the services considered on it. Section 6 indicates how to use the proposed architecture as a guideline to design these systems and also to evaluate already implemented platforms. Section 7 presents three experiences where the proposed architecture was used as design guideline during the development of these PVC supporting systems.

Section 8 reports three experiences where the architecture was used to evaluate already implemented platforms. Section 9 presents the conclusions and the future work.

2. Partially Virtual Communities

A partially virtual community (PVC) is a hybrid between a physical and a virtual community. This classification considers just the way in which their community members interact. Therefore, we assume that members of a 100% physical community perform just face-to-face interactions, and members of a 100% virtual community interact only through supporting systems (e.g. email or a Web application). Clearly, most communities involve physical and virtual interactions in varying percentages. The features of a hybrid community will be affected by the features of the physical and virtual communities, according to their percentages of representativeness. For example, a neighborhood community is a PVC that probably is closer to a 100% physical community, and a gamers community is a PVC that is probably closer to a 100% virtual community. In this article we consider PVC those communities that are in the middle area of this spectrum (Fig. 1).

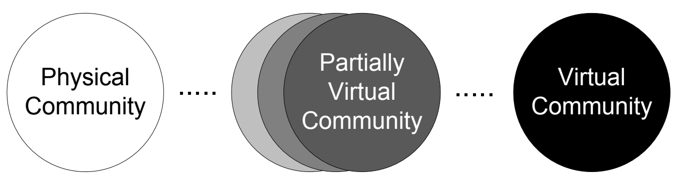

Figure 1. Spectrum of communities according to the nature of their member interaction

There is a lack of consensus regarding an appropriate definition of the terms physical and virtual community [Port04]. Therefore, for physical communities we adhere to the definition given by Ramsey and Beesley, which indicates that they correspond to a group of people who are bound together because they share the same living or work space, visit the same places, or otherwise spend a continuous portion of their time together [Rams07]. Regarding online communities, we adhere to the definition of Lee et al.

which indicates that they correspond to “a cyberspace supported by computer-based information technology, centered upon communication and interaction of participants to generate member-driven contents, resulting in a relationship being built up” [Lee03]. Based on these definitions, we define a PVC as *a group of people who interact around a shared interest or goal using technology-mediated and face-to-face mechanisms*. Depending on the community context, different PVCs could involve different degrees of *virtualness*.

In terms of size, PVCs accomplish with the "Dunbar's Number" [Dunb92] which indicates that human social networks involving stable relationships range between 100 and 200 individuals. These relationships are stable when an individual knows who each person is and also how each person relates to every other. This is true because according to [Gonc11], since both physical and virtual communities seem to already accomplish this.

Similar to physical and virtual communities, the PVC structure is diverse and may eventually become complex. The complexity comes from the fact that these communities could involve social and also (formal or informal) organizational goals. Therefore, the social structure that spontaneously rises through its member interaction is influenced by the organizational structure (in the case that this last one is present) generating a hybrid structure that is particular for each PVC community. However, we can assume a hierarchical structure for the PVC due it is basis of a social group [Chas80]. In fact, whenever a group of people interacts within a community, a leader-follower relationship almost always emerges [VanV99]. Therefore, we preliminary assume a leader-follower structure for a PVC, where it is possible to identify several roles, such as consumers, contributors, lurkers and veterans [Tedj05].

3. Related Work

This section presents an overview of the main requirements reported in the literature for systems supporting for activities in a PVC. It also presents and discusses the existing guidelines to model the PVC supporting platforms. Finally, it reports on specific dimensions to cover when assessing social support in these systems.

3.1 Requirements for PVC Supporting Systems

PVC platforms typically support information dissemination, self-service transactions, communication and mediation [Girg02]. The large amount of software to support online communities that exists today, may lead to misunderstand that the development of PVC platforms for particular purposes is straightforward [Chen05].

McMillan and Chavis [McMi86] state there are four elements that define a sense of community: *membership*, *influence*, *integration and fulfillment of needs*, and a *shared emotional connection*. Therefore, encouraging participation and allowing social interactions are basic requirements to be fulfilled by PVC platforms [West09].

People in a community tend to be similar to those which they have close connections with for two reasons: (1) they grow to resemble their current friends due to social influence [Frie98]; and (2) they tend to form new links to others who are already like them [McPh01]. While both factors are present in everyday social processes, they are in tension: social influence can push systems toward uniformity of behavior, while selection can lead to fragmentation. Crandall et al. found out social interaction is both an effect and a cause of selection, as there are clear feedback effects between these two factors [Cran08]. Moreover, online feedback mechanisms have emerged as a viable mechanism for fostering cooperation among strangers in electronic settings, by ensuring that the behavior of a certain user towards any other becomes publicly known and it may, therefore, affect the future behavior of the entire community towards that user [Dell03].

A persistent and updated identity triggers cooperation because community members tend to identify each other and keep a track of their past behavior [Koll96]. Moreover, the behavior and information published by the members under their personal profiles allow other community members to infer relationships and features of the participants [Misl10]. Lee et al. have identified a set of requirements that can be used to foster social interaction: *common ground*, *awareness*, *social interaction mechanisms* and *place-making* [Lee01]. Information sharing, knowledge of group activity and coordination are central to successful collaboration [Dour92]. Collaborative systems like PVC platforms should consider context to support interaction among group members. In fact, users of these systems especially value the information related to users' status and physical location, as well as their profile information [Hers12].

Although all these functional requirements (FR) identified in the literature are relevant in the design process of a PVC supporting system, establishing the non-functional requirements (NFR) is also highly relevant to obtain a design that helps keep the community alive. For example, scalability of these platforms is important since they usually provide support to several communities.

It is well known that the most effective way to address the NFR in a software system is considering them in its architectural design [Supa10] [Xu05]. Such an architecture must integrate harmoniously all FRs and NFRs of the system which, per se, is a challenge due to the interrelationships existing among these requirements. Moreover, the services provided by the architecture must be suitable for the end-users, particularly in PVCs where the members' interactions are based on a voluntary use of the supporting system.

3.2 Guidelines to Design PVC Supporting Systems

To the best of our knowledge, there are no particular proposals to help design the architecture of PVC supporting systems. However, there are some results from online community studies, which should be considered when modeling these systems. For example, Preece and Shneiderman [Pree09] have identified that community members are relatively shy at first, typically evolving from *readers* (passive stage) to *leaders* (active stage). Therefore, supporting services provided by a PVC platform must consider this user behavior evolution.

Hughed claims that large technological systems are complex and messy with no clear boundaries. Therefore, people within a technological system have a critical role, which is to complete a feedback loop, by perceiving the gap between system performance and system goals. In fact, it is only through this feedback loop that errors are caught and corrected, leading to improvement in system performance [Hugh89]. Kluger and DeNisi define feedback as the actions taken by external agents to provide information regarding some aspects of one's task [Klug96]. Tedjamulia et al. propose that performance feedback and social recognition are often used in online communities because (1) they are conical; (2) they allow self-organization of contributions in large sites; and (3) they bring to bear the collective, distributed, and significant human resources available in a community [Tedj05].

Beenen et al. claim that reminding community members of the uniqueness of their contributions, and setting individual or group goals for contribution, can enhance participation in a particular setting [Been04]. Similarly, Kim [Kim00] studied users in online communities and defined some guidelines, such as defining a community purpose, developing spaces for interaction, and creating meaningful profiles that may evolve in time. Porter [Port08] presents the *AOF Method* (activities, objects, features), which consists on a prioritization scheme for designing social Web applications, and a model of five stages of the usage lifecycle.

Gutierrez et al. state that participation is a key metric to evaluate the success of an online community [Guti11]. Based on that premise, they propose a framework for enabling interaction among users. The framework models virtual communities in three sections: (1) services that allow interaction, (2) participation and motivation strategies, and (3) definition of the software platform through which the community is going to interact.

Howard proposes a model to address the community member behavior and tries to identify the services required by them [Howa10]. This model is based on four components: *remuneration*, *influence*, *belonging* and *significance*.

Concerning guidelines for social platforms, Crumlish [Crum09] identifies a series of social interface design patterns and analyzes how they are applied into different systems. Van Duyne et al. [VanD06] present a pattern for designing online communities, considering policies, moderation, anonymity, interaction, trust, sociability, growth and sustainability. These patterns provide a partial solution to the design of PVCs, because they lack of support for physical interactions required by PVC members.

The literature also reports an ample variety of architectural and design patterns that were not particularly proposed to model PVCs, but could be used as general guidelines for it. For example, Schümmer and Lukosch define a pattern language for computer-mediated interaction [Schu07] that can be used to design several aspects of the community support, such as users identification, contacts (buddy list) and mechanisms for reciprocity and rewards among community members.

3.3 Evaluating PVC Supporting Systems

Concerning the evaluation of collaborative systems in general, Antunes et al. propose a framework to evaluate this kind of systems [Antu12]. This framework was conceived according to given variables and performance levels following the lifecycle stages along the system evolution.

One of the most important features to evaluate in software supporting social interaction is its *usability* [Pree01] [Pree03] [DeSo03]. In the field of Human-Computer Interaction, Nielsen [Niel93] defines usability in five criteria that need be satisfied: learnability, efficiency of use, memorability, few and non-catastrophic errors, and subjective satisfaction. On the other hand, these systems have to meet requirements linked to supporting social interaction (also called *sociability*) [Pree01] [Pree03] [DeSo03]. Among these factors, we can state governance structures such as moderation mechanisms, to the extent of establishing a common cultural context in the community.

4. Requirements to Support PVC Activities

This section identifies functional requirements (FR) and non-functional requirements (NFR) that are usually present in this type of supporting platforms. These requirements have been obtained from the literature review and the authors past experiences as designers of these software platforms.

Typically, PVC platforms are Web applications either open to public members or closed in private groups or organizations. The context that defines the community will state how information will flow outside its borders. For example, when the system must support inter-organization processes, interoperability should be considered as a mandatory requirement [Beng10].

These systems should implement at least two roles: *admin* and *standard* users. The admin-user takes the role of community manager, with permission to coordinate and control participation and membership. This is particularly important when participation is one of the expected outcomes of the activity. An administrator role contributes to keep the community governance within a certain suitability range and may be a way of responding to the perceived lack of strong governance structures in online communities [Para07] [Pree03].

In online communities, users generally need to express and expose their identities in a social context [Marc06]. For example, in social networking services such as Facebook or Google+, people manage their online presence through filling out user profiles. Moreover, we can infer the identity of a particular user by analyzing the content of his/her contributions and opinions [Misl10]. However, the disclosure of the online identity of a user may present a certain number of issues related to privacy [Born11] [Liu11] and the undesired disclosure of personal information.

Online reputation affects indeed the identity of a particular user in the context of an online community. Moreover, building and maintaining a good reputation can be a significant motivation for contributing to online communities [Guti11]. One strategy that communities tend to use to keep users aware of their level of reputation is monitoring the different actions that are currently being performed and those that have been performed in the past by the community members [Gart06] [Lamp07] [Lobr04].

When designing the interaction space, the supporting system should consider two disjointed environments: *public* and *private* [Neye11]. Sharing resources between these two environments should be possible. Public spaces foster communication throughout the community, and private spaces allow users to organize their personal information, as well as interact and share content with others. Moreover, social systems should include in their design feedback mechanisms for letting the information and current activities in the community flow between users and their interactions [Dwye08] [Kim00].

The platform architecture should also consider services that allow synchronous and asynchronous communication, as well as coordination mechanisms, among community members [Neye11]. It has to support three different kinds of interaction: user-to-user, user-to-a selected group and user-to-community. Counting on these strategies provides flexibility to user participation. Awareness about the members' availability usually helps to promote these interactions [Gutw98]. Since the community is partially physical, user location awareness mechanisms should be considered to trigger face-to-face interactions.

Concerning the NFR for PVC supporting systems, the most relevant and common ones seem to be: *performance*, *uptime, maintainability* and *scalability*. These requirements try to address the services usability (particularly the first two NFR) and the platform evolution. Other requirements such as *privacy* and *security* have also to be taken into consideration. Finally, in order to ensure member satisfaction towards the system, as well as effectiveness and efficiency when supporting user interaction, the software support has to comply with general *usability* principles. Table 1 summarizes the requirements to model of a PVC supporting system.

Table 1. Requirements for a PVC supporting system

Req.	**Description**
FR 01	The system should provide registration mechanisms that facilitate the appropriation of the platform by users [Crum09] [Howa10] [Kim00] [Marc06] [Port08].
FR 02	The system should provide mechanisms for managing a personal identity by users [Crum09] [Girg02] [Kim00] [Port08].
FR 03	The system should include awareness mechanisms in the form of users' availability, action identification and notifications [Crum09] [Dour92] [Fern06] [Hers12].

FR 04	The system may include location awareness in order to allow face-to-face interactions and break the barriers linked to virtualness [Crum09] [Hers12].
FR 05	The system should allow and trigger relationship building among community members; e.g. friends, circles, groups [Crum09] [Girg02].
FR 06	The system may provide services for sharing content and media with other users, either in private groups or publicly [Crum09] [Guti11].
FR 07	The system should provide interaction mechanisms, such as synchronous and asynchronous communication modules [Crum09] [Kim00] [Neye11] [VanD06].
FR08	The system should provide appropriate feedback to its users on the different actions performed by them [Dwye08] [Kim00].
FR 09	The system may provide mechanisms for supporting coordination, such as creating content collaboratively, among community members [Crum09] [Guti11].
FR 10	The system should include control mechanisms, such as peer moderation, governance structures and filters [Crum09] [Para07] [Pree03] [VanD06].
FR 11	The system should be designed following a motivation and participation strategy in order to ensure a certain level of activity through time [Chen05] [Girg02] [Guti11] [Pree09].
FR12	The system should allow monitoring user activity and contributions performed in the software support through time [Gart06] [Lamp07] [Lobr04].
NFR 01	The system should quickly react to any request made by users or its components [Mena01].

NFR 02	The system should be highly available (uptime), since PVCs are supposed to break down time barriers, allowing members to interact at any time [Mena01].
NFR 03	The system should be maintainable and extensible, because communities evolve naturally in time and follow a specific lifecycle, as well as its users [Busc96].
NFR 04	The system should be scalable, since it has to be able to handle a continuous growing number of users and contributions made within the community [Hill09].
NFR 05	The system should ensure privacy and security, as PVCs have to be trustworthy for users in order to trigger interactions [Born11] [Crum09] [Guti11] [Liu11].
NFR 06	The system should be usable, since it has to support community member interaction and deal with different kinds of users [Guti11] [Pree01].

5. Software Architecture for PVC Supporting Systems

Herskovic et al. state that, according to the separation of concerns necessary to face the *iceberg effect*, the requirements of collaboration systems should be layered: requirements in the upper layers are highly visible to users and developers because they represent services that are exposed to end-users through the application front-end [Hers12]. Following this line of reasoning, we propose a software architecture composed of three layers (Fig. 2): *user*, *interaction* and *community* layer. The *User Layer* refers to specific actions to be performed by a single user within the community. Some of the expected tasks to be carried out by a user are logging into the software and managing his/her profile and personal identity. The *Interaction Layer* refers to all actions and services to be done by two or more users, or with the intention of causing an effect on the community. The *Community Layer* refers to the global scope of the community, the elements that define the software, and all the principles that directly affect the whole group.

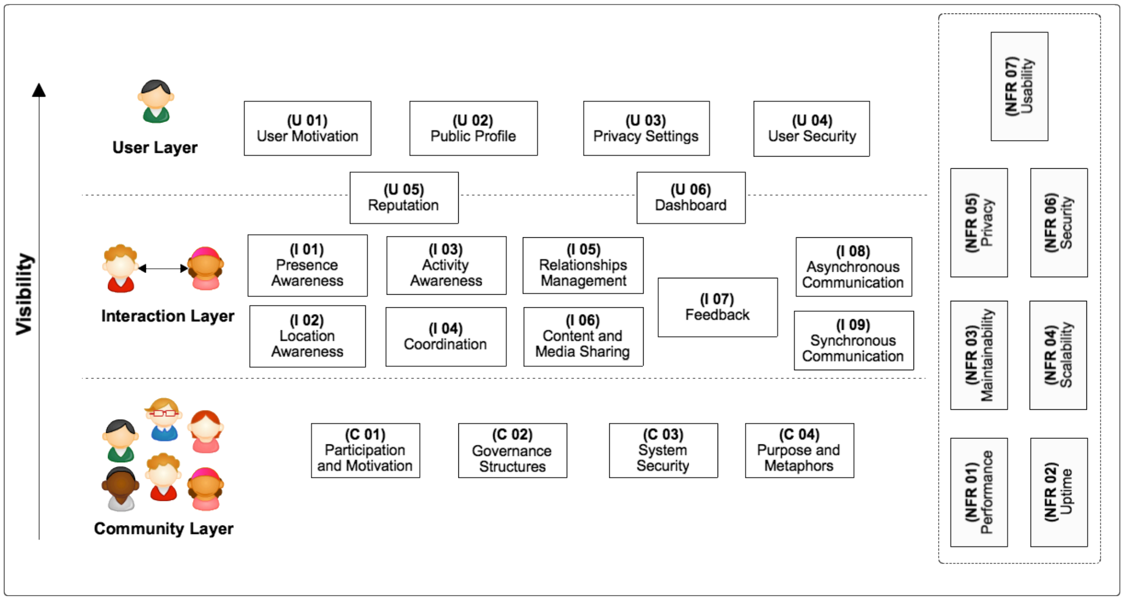

Figure 2. Software architecture for PVC supporting systems

The User Layer is composed of six services; two of which are shared with the Interaction Layer. The *user motivation*, *public profile*, *privacy settings* and *user security* manage the identity and visibility of a single community member in the software platform. The *dashboard* is where personal contributions are published alongside those of the other members. It allows filling-up the *feedback loop* of information where personal and public notifications foster interaction among users. The *reputation* mechanisms act as a visible input of this feedback loop and may trigger and enhance participation and interaction among users.

The Interaction Layer is composed of nine services: *presence awareness*, *location awareness*, *activity* awareness, coordination, *relationships management*, *content and media sharing*, *feedback*, and *synchronous and asynchronous communication*. The two requirements related to awareness are justified because of the need of users to foster face-to-face interactions, as well as requirements linked to services providing different communication channels for users interaction, e.g. a message board or a chat room. The relationships management component is a key issue in this architecture. Such a service allows users to identify other members and send an interaction request to them. The coordination service regulates the access to shared resources of the community (e.g. shared object or the communication channel). The content and media sharing component is closely linked to participation in communities that are based on collaborative work. Using such a service, users may interact with each other to contribute or create new content, thus leading the community to evolve.

In the Community Layer we can identify the four mechanisms (rather than proper software services) that define the context where a community lives and evolves in time. These mechanisms are: *participation and motivation* strategies, *governance* structures, *activity monitoring*, and the *purpose* and linked *metaphors* to be used when designing the community. In particular, this layer is usually invisible to end-users, because its components affect the whole structure of a community. However, it is the one that has the greatest impact in the design of PVC supporting systems. These services act as a whole structuring the community on how its members are going to participate and contribute, under which kind of cultural environment, in which context the community is going to be sustained and how user activity is going to be monitored along the community lifecycle.

Concerning the NFRs, they are "transversal requirements"; therefore, they concern all the services provided through the architecture. The proposed architecture considers these NFR and proposes mechanisms to address them. Particularly, the identification of services and their separation by concerns (i.e. user, interaction and community) make the systems *maintainable* and *extensible*. This property comes from structuring the systems using layers [Busc96]. We can also expect an appropriate *performance* of the systems that are implemented using this architecture because it is client-server and involves just three layers [Mena01]. Since the two lower layers (which are affected by the number of communities and users to be supported) live in the server, we can ensure the system *scalability* by increasing the computing power at server side. The system *uptime* cannot be ensured through this architecture since it does not consider replicated components in the server side [Mena01]. It should be interesting to include this requirement it in the future. However, the proposed architecture partially addresses such a NFR through the use of asynchronous interaction services.

User *privacy* preferences are stored by the system; therefore the services provided by the platform must self-configure to adhere to the user privacy settings. Since this information is kept in a dual-synchronized way (i.e. at client as well as at server side), it cannot be modified unless the user has simultaneous access to both copies of such information. This information management policy is also used to manage the personal and login information. This mechanism contributes to build *secure* systems. In addition, the architecture considers users authentication. Similar to any other domain specific software architecture, this proposal addresses the systems *usability* just accomplishing with all previous requirements (including FR and NFR).

The complexity of the architecture presented in Fig. 2 and the nature of these supporting applications indicate that these systems must be framed in a client-server architecture, where the user layer runs at client side, and the two lower layers at server side. This design decision simplifies the services implementation. Table 2 shows the requirements traceability matrix, matching the requirements defined in section 4 and the different components of the proposed architecture in Fig. 2. Since NFR are transversal to the three layers, they are not depicted in the table.

Table 2. Requirements traceability matrix

	FR01	FR02	FR03	FR04	FR05	FR06	FR07	FR08	FR09	FR10	FR11	FR12
U01	X	X										
U02		X										
U03		X										X
U04		X										X
U05		X						X		X	X	
U06		X						X			X	
I01			X									
I02				X								
I03			X									
I04									X			
I05					X							
I06						X						
I07								X				
I08							X					
I09							X					
C01											X	
C02										X		
C03												X
C04											X	

6. Using the Proposed Architecture

The proposed software architecture involves three layers (i.e. the user layer, interaction layer and community layer) and also a transversal set of NFR that affects the services of the whole architecture. Following the *iceberg metaphor* introduced by Herskovic et al. [Hers12], the services in the upper layer represent the visible area of the iceberg, because they are more concrete and visible for designers. Contrarily, services in the lower layer are more abstract and less visible for designers; therefore they are the hidden part of the iceberg. Considering this structure, every new design of a PVC supporting system must consider at least this separation of concerns, and analyze the need to include the proposed services following a top-down iterative strategy. This strategy allows designers to address services from those that are more concrete and simple to those more complex and abstract. This strategy represents the natural way to deal with the challenges. The NFR layer, which is also part of the non-visible area of the iceberg, is used as an evaluation element that allows designers to determine if a design is stable and robust.

Next subsections describe how to use this architecture as a guideline for the development process, and also as an instrument to evaluate already implemented platforms. Figure 3 summarizes the activities involved in these processes when the proposed architecture is used as a guideline or an instrument respectively.

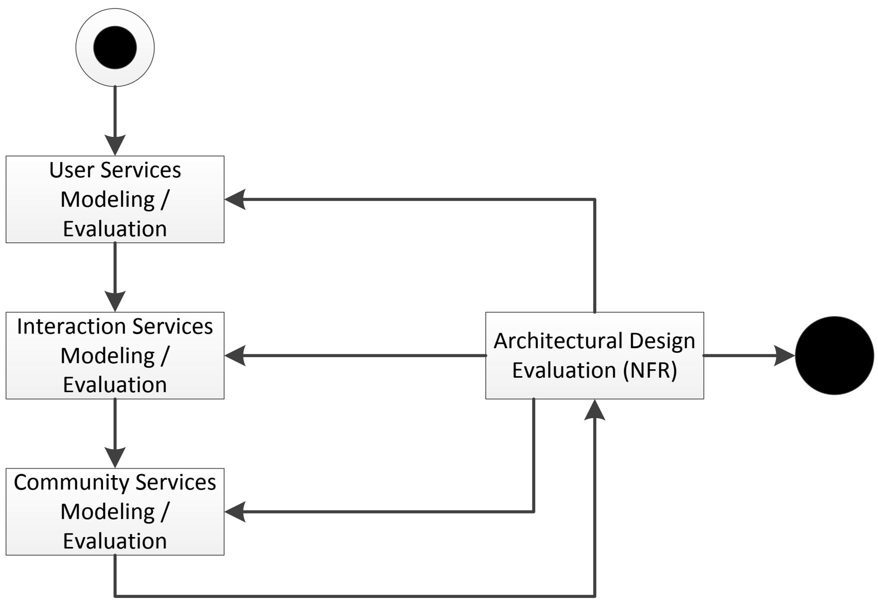

Figure 3. Process to use the proposed architecture

6.1. Developing a PVC Supporting System

As mentioned in Section 5, it is recommended that the architectural design of a PVC supporting system considers (at least) the separation of concerns shown in the proposed architecture, because it contributes to enhance the maintainability, extensibility, performance, scalability and uptime of the implemented solution. Once established this basic structure for the new system, the designer must analyze the pertinence of the services considered in the proposed architecture, following a top-down and iterative manner, starting with the *user services* (upper layer in Fig. 2). The goal of this activity is to determine, based on the user requirements of the project, if a certain service is critical (i.e. required), desirable or non-required for the new product. In order to determine the level of requirements coverage provided by the selected services, it is recommended to do a traceability matrix that matches the functional (FR) and non-functional requirements (NFR) to the selected services. Thus, the designers can identify which requirements cannot be addressed with the services proposed in the architecture; therefore they must then work in the design of a particular solution to deal with those requirements.

For each requirement considered as critical or desirable, the designers must propose a particular design that captures the motivation and need of the community members. For services in the *user* and *interaction* layers, such a design must include a user interface. Services in the community layer usually do not require a user interface; however the designers have to decide about it.

This analysis and modeling process must be done sequentially (i.e. layer by layer) and top-down. After considering the community layer, the designers will count on a layered architecture that embeds services required to support the activities of the PVC community. Such an architecture must be then evaluated considering the non-functional requirements that are relevant for that project. Typically this evaluation is done using an *inspection*, which is a type of formal technical review [Free00]. These evaluations help designers to see if the obtained design is stable and also if it addresses the FR and NFR identified in that project. If the design is deficient, this activity provides the feedback required to adjust the structural model or the services composition in order to obtain a stable and robust design. This process is done iteratively until getting a strong design.

6.2. Evaluating an Already Implemented PVC Supporting System

The process that designers must follow to use of the architecture as an instrument to evaluate already implemented PVC supporting systems, is similar to the one presented in the previous section. In this scenario the evaluators can count or not with the source code of the platform being evaluated. In the first case (i.e. if the evaluators have access to the source code), the process is the similar to the one described in section 6.1, but the product being evaluated is the implemented system instead of its design. If the evaluators have the product's source code and its design, the evaluation is done using both products; therefore the eventual adjustment process (that is done based on the evaluation feedback) should also be done on both software artifacts. However, if the evaluators count just with the source code, they have then to perform a simple reverse engineering process to get an approximation to the product's design. That design allows them to get a feeling about how well the implemented system addresses the NFR that are relevant for the community. Clearly, more design information of the product will allow more accurate evaluations.

In the second case (i.e. if the evaluators do not count on the source code), it is not possible to modify the product design or its implementation; however the evaluators can determine if an already implemented system is suitable to support the activities of a certain community. This evaluation type can be used to create a ranking with the candidate systems for a community, and also to determine which critical services are not supported by each platform. Due that today there is an important number of software platforms available for free (e.g. the social networking services), this evaluation capability of the proposed architecture could contribute to identify the best supporting system for a particular PVC.

7. Designing PVC Supporting Systems

In this section we report three experiences that indicate how the proposed architecture was used to support the design of the software platforms for three particular PVCs. The first two case studies were conducted by groups of advanced undergraduate students during the *Social Computing* course (CC5703) between August and November of 2012. This course, which is delivered by the Computer Science Department of the University of Chile, involved 16 weeks. During such a time period, the students have to conceive, design and preliminarily validate a software platform that supports the activities of already existent PVCs. The third case study describes the development and evaluation of a platform that supports activities of a course community at the Business School of the University of Chile.

In these three case studies the authors monitored the development process, the product evolution and the usefulness of the proposed architecture. The monitoring process involved periodical meetings with the students, focus groups and also formal technical reviews. Next subsections describe these experiences and the obtained results.

7.1. Supporting a Bikers Community

The group of students that addressed this project noted that the bikers community of the Engineering School at the University of Chile lacked of proper mechanisms for managing their community members, supporting their interactions, and managing and advertising the events that they periodically organize. Their project main goal was to design a PVC supporting platform for that community, highlighting the events organization and also enhancing the interactions among community members around these events.

The proposed solution was designed thinking in a smartphone as the target device. Figure 4 shows the main user interface of such a PVC supporting platform. Users of this mobile application can post events and advertise them within their groups and also through other social networking sites, such as Facebook and Twitter. Every member has a profile, a reputation, a privacy setting and one or more friend lists. The main user interface shows the public messages in a dashboard, provides several awareness mechanisms, and presents several communication and coordination tools to the users.

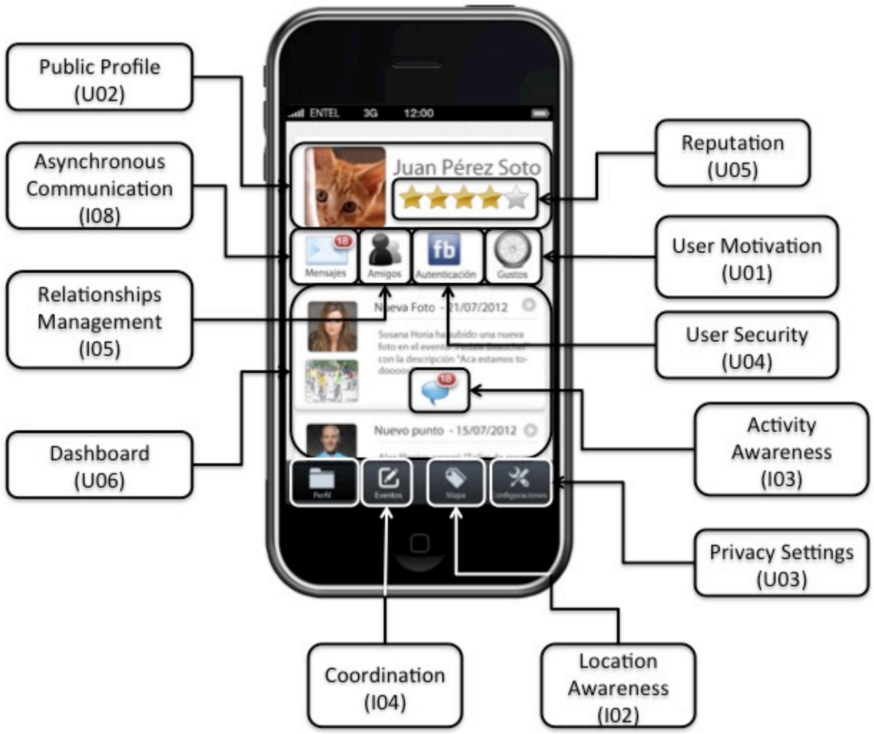

Figure 4. Main user interface of the bikers' community mobile application

Due to the fact the design of this application was part of students work in a formal Computer Science course, it involved an incremental process with five stages: (1) to research, conceptualize and characterize the community to be supported, and also the problem to be addressed in the project; (2) to establish the user requirements to be considered and perform a preliminary design of the supporting platform; (3) to perform an architectural design of the platform; (4) to build a detailed design of the supporting system; and (5) to present a functional prototype of the designed platform. A milestone was established at the end of each stage to determine if a team could advance to the next stage. Each milestone involved a deliverable and a Formal Technical Review (FTR) on such a product. The FTR followed the dynamic established for an inspection [Free00] and each of them involved 50 minutes approximately per project. In these sessions, the inspectors (i.e. the instructor, the teaching assistant and the rest of students) used the proposed architecture as a guideline to help the students to improve the design of their solutions. Every design aspect identified in the proposed architecture was discussed and evaluated in the last three stages of the design process. The feedback provided by the inspectors was recorded, and then addressed by the designers. During the next checkpoint the students should indicate how such a feedback was considered in the new design of the product.

The last stage of the project involved the development of an application prototype composed of user interfaces (mock-ups) and its navigation. The prototype allows a user to evaluate the proposal as if it were an already implemented product. Using such a prototype, the instructor and the teaching assistant evaluated the functionality of the platform and how well the different design aspects of the product were addressed.

The use of the proposed architecture helped the students to identify design aspects of the product that were not identified initially by them, as for example the need to include location awareness mechanisms and privacy settings. The design of this PVC supporting platform was graded with a 6.5 (in a range from 1 to 7). Four design aspects were not included in the prototype; however all of them were opportunely evaluated. The *presence awareness* (I01, see Fig. 2) and *synchronous communication* support (I09) were not included because the designers decided to just use asynchronous communication as a way to keep persistent interactions among the community members. The other two design aspects that were not included are *feedback* (I07) and *content and media sharing* (I06). This represents a deficiency in the platform design, which was not addressed by the students because of time reasons. The design aspects recommended for the community layer (see Fig. 2) were considered in the platform design, but they are not visible through the application user interface.

7.2. Supporting a Players Community

The participants involved in this project found out that the players of card, board and role-playing games have many common interests. Groups of these players regularly organize meetings and tournaments, and they are part of a players' community. The supporting platform designed in this project should enhance the sense of community among these players, help them to organize events and promote them among other players communities. The platform was designed thinking in a laptop/desktop as the target device. Figure 5 shows the main user interface of the proposed solution.

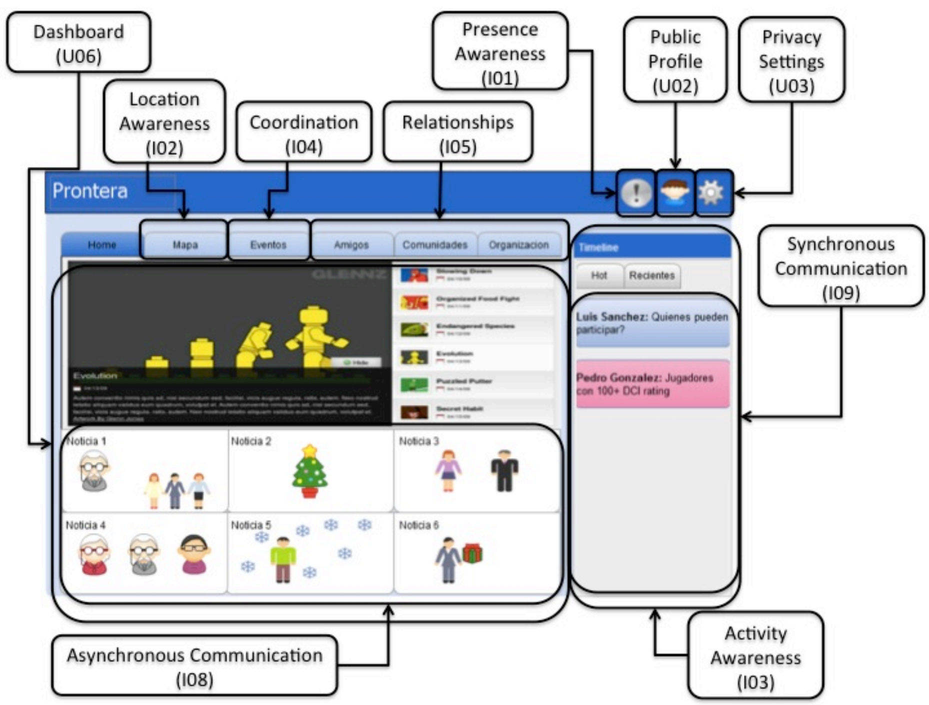

Figure 5. Main user interface of the players' community

Since this project was also developed in the context of the *Social Computing* course (CC5703), it follows the same dynamics explained for the previous case study. This platform embedded all design aspects specified in the proposed architecture, but it failed implementing some of them; for example, the *activity awareness* (I03) was implemented using a chat, and the *users presence awareness* (I01) was not permanently visible in the platform. This design was graded with a 6.5 (in a scale from 1 to 7), and the students agreed on the limitations of this design to represent some of the proposed aspects. After performing the two reported projects, no additional design aspects were identified as potential extensions to the proposed architecture.

7.3. Supporting a Course Community

In order to determine if the services considered in the proposed software architecture are suitable to support the activities of a real PVC, we have developed the supporting system for an existing community. This community was composed of thirty students of an introductory *Information Technology* course of the Business School at the University of Chile. Students taking part of this experience were volunteers and were required to register and validate their accounts. They were also asked to fill up their personal spaces and publish, rate and comment discussion topics related to the course contents.

The lecturer and two teaching assistants also became community members and discussed with the students. The users participated through an avatar and a pseudonym to keep anonymous their interactions. The community had a manager (an external user) who tracked these interactions and gave regularly feedback to members about their participation in this virtual scenario. The tracking period was limited to 8 weeks from its initial launch, between March and May 2012. After that period we applied a survey to end-users to gather their opinion about the usefulness of the services provided by the platform, which considered all design aspect specified in the proposed architecture.

After such a validation process we carried out a focus group with six software designers: two with experience in the design of social platforms, two with experience evaluating usability of software interfaces, and the last two with no prior knowledge about modeling PVC supporting platforms. The focus group served to discuss and clarify the designers' opinion about the suitability of services and pertinence of the NFR considered in the proposal. Each designer filled up the survey and a section asking for the suitability of the considered NFR. Using these results we tried to answer the following questions:

(Q1) Are the services considered in the architecture useful to support the interactions among members of a PVC?

(Q2) Is the architecture a guide to design PVC supporting systems?

(Q3) Is the architecture useful to evaluate already implemented PVC supporting platforms?

Next we briefly describe the design and implementation of this PVC supporting platform (section 7.3.1). Then, we present the survey (section 7.3.2) and the results obtained in the experimentation process with the users of the PVC supporting platform (section 7.3.3). Finally, we report the results gathered in the focus group with software designers (section 7.3.4).

7.3.1 Design and implementation of the supporting platform

This platform was implemented as a desktop Web application, by modifying and extending the core functionalities offered by the PHP framework *Elgg* [Elgg12]. Figure 6 shows the main user interface of the platform, where we can identify components related to most of the design aspects considered in the proposed architecture. This experimental prototype was designed for testing the completeness and suitability of services considered in the proposed architecture. Particularly we aimed the tool as an extra support for the discussions of the topics treated in each lecture.

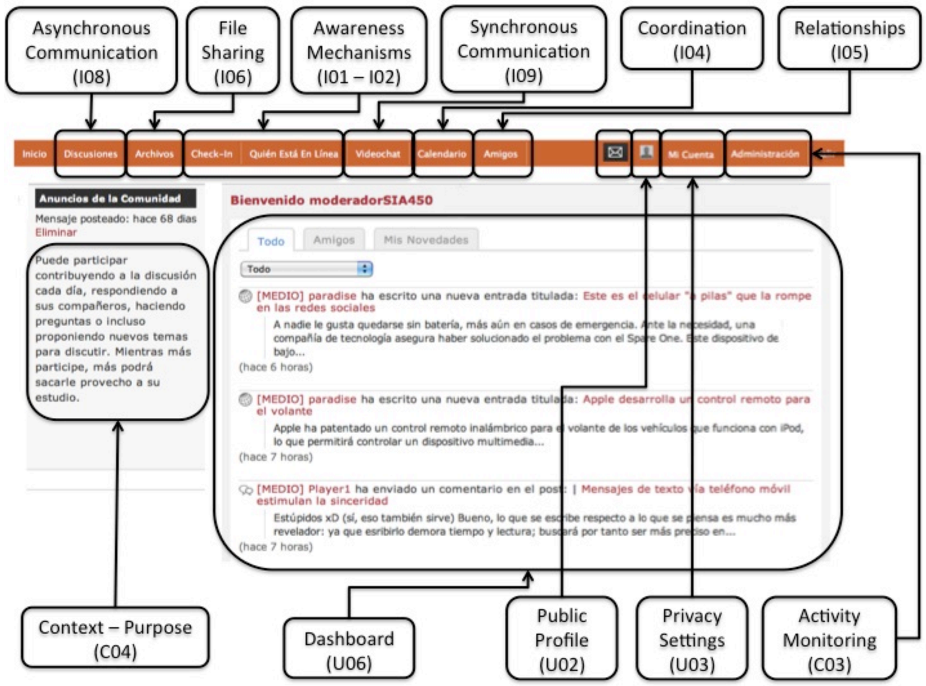

Figure 6. Main user interface of the developed platform

As a participation strategy, we developed an algorithm that classifies users according to their participation level. This information is displayed through a public label embedded next to each contribution and comment made by a user. These labels were *high, medium* and *low*, according to the perceived value (i.e. quantity and quality) of the user contributions. Users can evaluate the contributions of other community members using a scale between 0 and 5.

In order to model a function that quantifies the participation of each user, we considered a set of predefined tasks to be executed by the community members. Each task was affected by a weight factor according to the relative importance that it has in fostering interactions in the community. Therefore, the factor affecting complex tasks (such as commenting an article or contributing with a discussion topic) will be higher than those affecting simple tasks (such as logging in the platform). The participation score of each user corresponds to the linear combination of the score associated to each item and the weight factor of each task.

We also define the *quality factor* as the weighted mean value of the rating that community members assign to the contributions. Finally, in order to motivate the use of the platform (and the quality of them), the participation value is decreased in 15% per week. This strategy is inspired from the proposal of Cheng and Vassileva [Chen05].

Every week, we ranked the users according to their participation score and assigned them to the different categories in order to help improve their participation in the community. The first classification assigned a 20% of users in the *low* group, 20% in the *high* group, and a 60% in the *medium* group. The reason for this distribution lays on the assumption that users affected in the lowest category may feel the peer pressure of the public label linked to his/her username in the community; on the other hands, users affected in the highest category might feel the risk of decreasing their recognition status if they do not participate enough [Chen05].

Finally, we introduced a *security band* where users may stay if their category is affected in no more than a 5% of the participation score in the limits of each category [Guti11]. This helps prevent frustration in users that might see their category decreased when their scores are close to the limits between the low-medium and medium-high categories.

7.3.2 Survey

The survey included an item for each service of the proposed architecture. Users graded the usefulness of those services using a 5-point Likert scale. Values of 1 and 2 correspond to “negligible” services, a value of 3 corresponds to a “desirable” service, and a value of 4 or 5 means that the service is “mandatory”. The survey also included a section for open comments where the users could suggest services to improve the supporting platform.

Some services considered in the model, such as the user security settings were not considered in the survey, since they are either used only once or required just to access the supporting platform. Similarly, the user motivation, feedback loops, activity monitoring, governance structures, motivation and participation, and purpose and metaphors were also left out because they are invisible to end-users. Table 2 summarizes the services considered in the survey.

Table 2. Description of supporting services

Service	Description
Public Profile (U02)	Users have a personal space where they can manage their virtual identity. It provides support for an avatar, personal status or interests.
Privacy Settings (U03)	Users can decide what information will remain public and private. Also, they manage how they will receive notifications (e.g. email, in-site).
Dashboard (U06)	A main page where is published automatically the recent activity in the community, such as new messages and recent contributions.
Presence Awareness (I01)	Users can see the list of the other community members that are currently logged-in into the platform.
Location Awareness (I02)	Users can indicate their location by choosing a place from a list of options. If there are two users at the same place and time, they will receive a notification according to their privacy settings.
Coordination (I04)	The system provides a calendar with different permission levels: users can schedule activities that are private, or involve groups.
Relationships Management (I05)	Users can specify relationships among them, such as being part of a same group or being friends. This requires symmetric validation.
Content and Media Sharing (I06)	The system supports media uploading (e.g. documents, pictures and videos), classifies it into categories and allows users to comment on them.

Asynchronous Communication (I08)	Users can publish, comment and rate discussions related to the different topics they have worked on the lecture sessions.
Synchronous Communication (I09)	The platform supports a video chat room for logged-in users. They have to allow camera and microphone access beforehand.

7.3.3 Users Perception versus Designers Perception

Figure 7 shows the usefulness of each proposed service according to users and designers. Dark bars represent the average value assigned by the users to the *services usefulness*. Light bars indicate the *perceived usefulness* of each service according to designers' opinion. A continue scale from 0 to 10 was used to represent the usefulness of each service.

The results indicate that most services were useful for the community members. Moreover, the usefulness assigned by the end-users was similar to the ones assigned by the software designers. Analyzing the results and also the students' comments in the survey, we have identified some problems in the services implementation. Services like synchronous communication and coordination were not suitably implemented in the PVC supporting system. Therefore there is an important gap between the expected and the perceived value of such services.

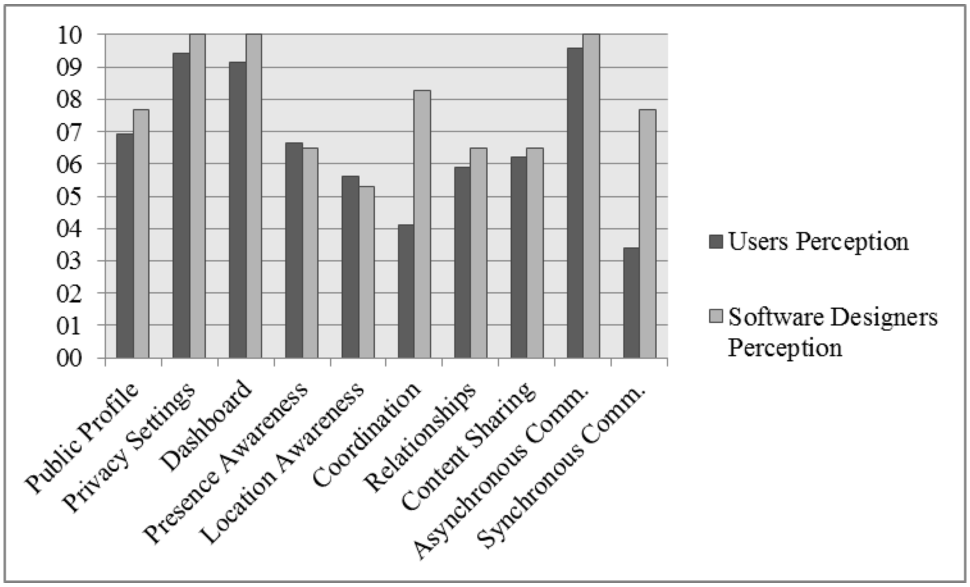

Figure 7. Services usefulness perception according to users and software designers

The spontaneous responses given by five end-users into the survey indicate that they would have preferred a simple chat room instead of the video-chat embedded in the system. This reflects that the community in fact requires this service, but it was not implemented properly.

Concerning the coordination service, the panel of software designers recognized the importance to count on this service, but the end-users assigned a usefulness value considerably lower than expected. This was also reflected on spontaneous comments that end-users stated at the end of the survey. These comments show a lack of initiative of the students to use such a service, since it was not required to perform the community activities during the experimentation period. If the use of this service would be required for either the community manager or other users to perform some activity, then the community members would have recognized its value.

Summarizing, these results show that all services considered in the software architecture are useful according to the end-users. This preliminary conclusion provides a first response to the Q1. In case of identifying a gap between the reported and the expected usefulness of a service, the cause can be: (1) inappropriate service implementation (that is the case of the synchronous communication mechanism), and (2) lack of initiative (or need) for using such a service (that is the case of the coordination service).

7.3.4 Focus Group with Software Designers

According to the designers' opinion, the FR and NFR were appropriately considered in the design of the PVC supporting system. These engineers also highlighted the simplicity of the software architecture, which make it usable for many people. They were able to quickly understand the separation of concerns represented by the three layers architecture.

Five participants spontaneously pointed out that the designed platform lacked of support for activity awareness, as this feature would enhance participation between community members. This observation provides a preliminary response to Q2, which is also supported by the results shown in Figure 7. All designers considered this architecture to be useful for analyzing other services in different contexts, and also easy to learn. Moreover, they think that the architecture could be used to evaluate already implemented PVC supporting systems. This provides a preliminary response to Q3.

8. Evaluating PVC Supporting Systems

In order to show how the proposed architecture can be applied in practice, we will briefly analyze three commercial PVC supporting systems: *Facebook*, *U-Cursos* and *AcaMed*. In this analysis we attempt to verify whether or not these systems satisfy the set of requirements specified in section 4, and also if the non-addressed requirements are effectively required by the community members. Thus, we intend to show that this proposal can also be used for: (1) choosing an already implemented system from a set of possibilities, and (2) identifying further customizations or extensions that have to be included in a supporting system that is currently being used by a specific community.

8.1 Facebook

Facebook is considered as one of the most successful social platforms. Although this is a general social system, it can be used to support PVC with the *Groups* feature. A *Facebook Group* offers the same services as Facebook, but restricted to a particular group of users. Membership, visibility and moderation of these groups are supported by one or more *group admins*, and *standard users* are linked together through their own Facebook profiles. Figure 8 shows the main user interface of the product, and it identifies the components that match with the proposed architecture.

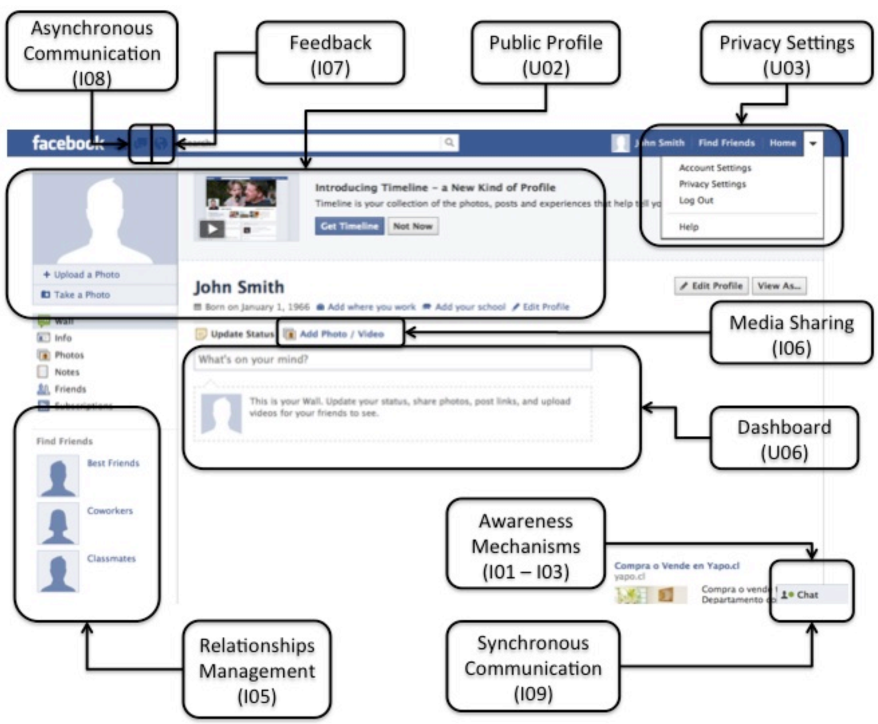

Figure 8. Facebook Web interface for a user

We can see most services considered in the architecture are part of Facebook. However, this platform lacks of support for coordination mechanisms (e.g. community agenda or community members commitments), which can be a consequence of a design decision. Facebook currently supports location awareness in form of geo-tagging in status messages and in personal and shared pictures. This location awareness implementation limits the activities of a PVCs because it does not provide mechanisms for coordinating people, and thus promoting physical encounters. This limitation is not surprising because Facebook was not particularly designed to support PVCs. The identification of such a limitation allows us to show that the proposed architecture can be used as a reference to identify mandatory services in PVC supporting systems.

Services considered in the community layer were not identified in this evaluation process, because they are not visible for regular users. Something similar occurred with the evaluation of the system architecture and the strategy used to address the NFR.

8.2 U-Cursos

U-Cursos[1] is a PVC supporting system developed at the University of Chile for managing courses and fostering interaction among courses participants: lecturers, teaching assistants and students. Currently, this platform is commercial. In the system, each course defines a specific context in the form of an independent community. Interaction is achieved through asynchronous communication (email and a discussion forum), and community members may upload and download class material and related media content. Figure 9 shows the main user interface of *U-Cursos*.

[1] https://www.u-cursos.cl/

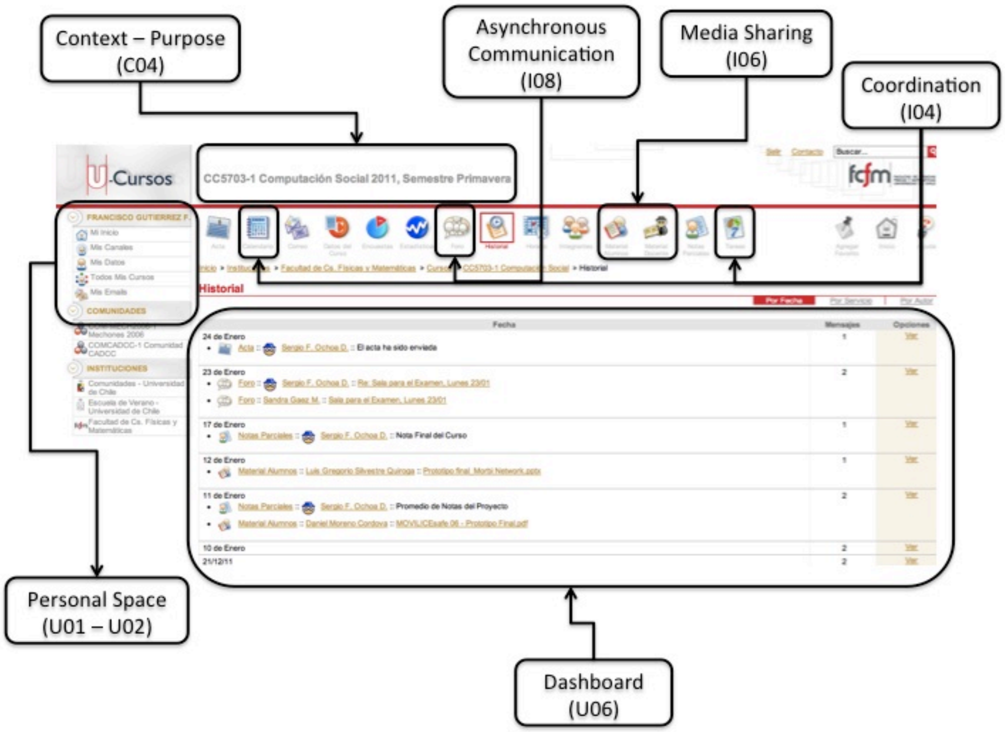

Figure 9. U-Cursos main user interface

The *U-Cursos* limitations come from the system conception. This tool was not initially designed to support PVCs, but it was evolving over time up to a tool that plays such role. Therefore, the required support for the community members' activities is still incomplete. For example, the system lacks of services that stimulate interaction between users. Moreover, there is not a proper participation strategy that would eventually transform this information system into a proper PVC supporting system. The platform includes several coordination services, but it still does not support location or presence awareness.

Since the authors are regular users of this platform, we can confirm the need to count on the previously mentioned services. These limitations have also been discussed with the engineers in charge of this platform evolution, who agree that the mentioned services must be included in the system. Hypothetically, if the *U-Cursos* design were based on the proposed architecture, the implemented and also the pending services would be identified in an early stage of the system development.

The services provided by *U-Cursos* in the *community* layer and also its software structure and quality features were not available for the authors during the evaluation period; therefore, they were not considered in this analysis. In other words, we just considered in this evaluation just services in the visible part of the iceberg.

8.3 AcaMed

The *AcaMed* website is intended for creating communities of health professionals in Japan. It can be considered as a meta-social platform since it provides free tools for launching websites for a certain medical society. It also allows managing members, membership fees and customizing the society's website by activating various services that its members can use. Currently 600 medical societies in Japan are using this platform. Some of the services that it provides are the following:

- Publishing news for the society members.
- Coordinating and delivering online conferences.
- Sharing data and reports about rare medical cases.
- Sharing multimedia content like videos, presentations and medical images.
- Managing the paper submission process for conferences that the societies organize.

Figure 10 shows the main page of a website created using *AcaMed*, for a conference organized by the pediatric surgery society.

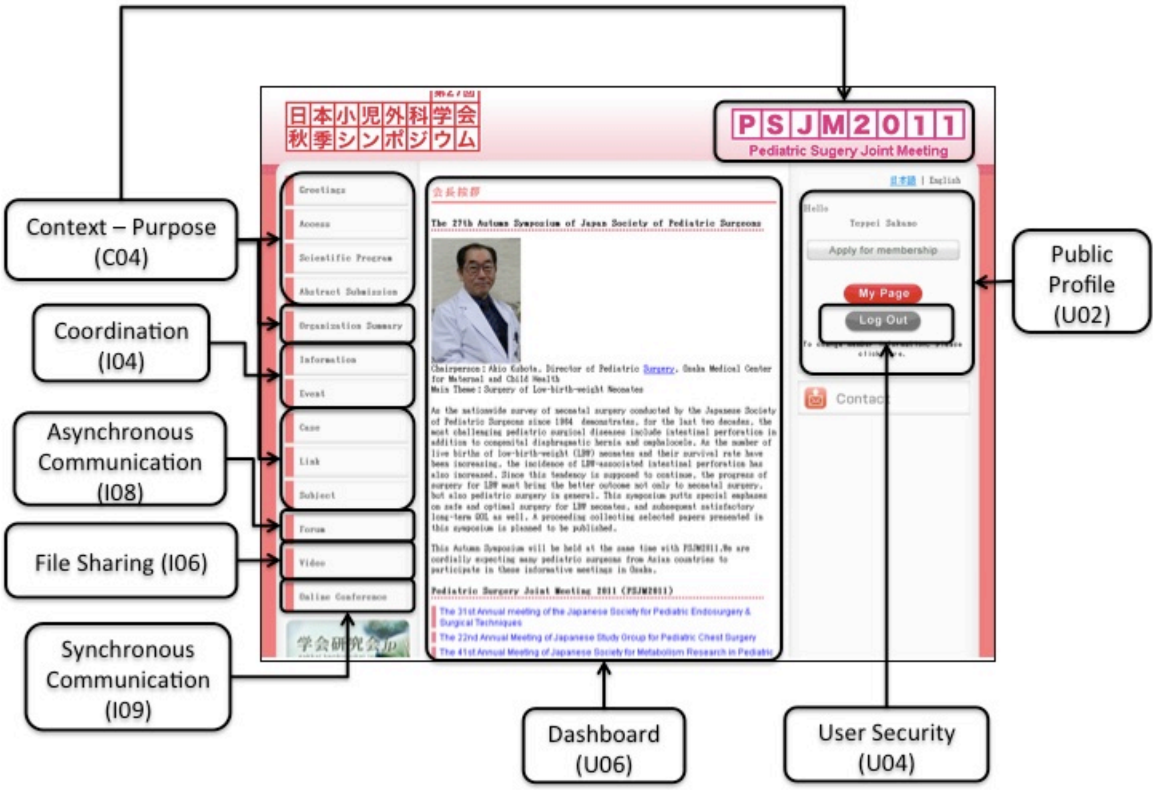

Figure 10. Main page of a website created using AcaMed

According to the managers of this supporting platform, its success relays mainly on the flexibility that it has to adapt the website not only to the particular needs of a certain society, but also to evolve this support as the community itself evolves. When confronting the services available in *AcaMed* with those considered in the proposed architecture we can see that most of them are implemented. Therefore, the users that maintain a website for a certain society may include or take out most of these functionalities even dynamically during the lifecycle of the community. The non-implemented functionalities in this platform were *user motivation*, *public profile*, *user security*, *presence* and *activity awareness*, and *location awareness*. When we asked about the possibility to include these missing features the website developers answered it might be a good idea to include a *user motivation* mechanism, for example in the form of an automatic mail service, which will alert users when important information is uploaded to the website. Regarding the inclusion of a *public profile,* they considered that this is definitely a good idea, which was missing in the current implementation. About *security mechanisms* they said that the data managed by the system is not sensible enough to consider the implementation of such functionality as a priority. They also considered the introduction of *presence* and *activity awareness* not important for these communities, since most interaction among members occurs asynchronously, except for the online conferences where a basic awareness mechanism has been implemented. The inclusion of current *location awareness* was considered not important, contrary to the *permanent location awareness*. In this kind of network it would be important to know where the permanent location of a certain specialist is. This feature might be implemented along with the *public profile* availability. In conclusion, we can say that the application of the framework for evaluating this PVC supporting platform helped to identify its possible limitations and extensions.

9. Conclusions and Future Work

According to Gutierrez et al. [Guti12] a Partially Virtual Community is a hybrid between a physical and a virtual community, and it can defined as *a group of people who interact around a shared interest or goal using technology-mediated and face-to-face mechanisms*. This article proposes a software architecture that helps design PVC supporting systems and also evaluate already implemented platforms. This architecture considers several recurrent services that have been reported in the literature and also identified by the authors during the development and evaluation of PVC supporting systems. These services have been arranged in a layered structure that considers a horizontal and a vertical dimension. The horizontal dimension (i.e. the virtualness) indicates if a certain service is usually required by a physical community, by a virtual community or by both of them. The vertical dimension (i.e. visibility) separates the services by concerns (e.g. services for users, to support interactions and for the community), indicating the visibility level of them. The services visibility decreases with the layer in which such a service is located. The layered feature of the proposed architecture contributes to enhance the maintainability, extensibility, performance, scalability and uptime of the implemented. These capabilities are important for PVC supporting systems because they have to show a high availability and also evolve almost constantly.

The article reports how this architecture was used to design three PVC supporting systems, and evaluate three already implemented platforms. The obtained results were analyzed according to the research questions stated for this research work. Answering these questions will require evaluating the proposal more in-depth; however, the preliminary results indicate that the proposed software architecture considers services that are useful to support interactions among members of a PVC (*Q1*). The architecture would also be useful to support the design of these systems (*Q2*) and the evaluation already implemented platforms (*Q3*).

The next steps in this research work consider evaluating the supporting platforms of the reported communities, but now counting on the help of their designers. This activity will provide us a better understanding of the strengths and weaknesses of this proposal, because we can address completely all layers considered in the architecture. Moreover we will intend to determine if the list of services considered in the proposed architecture is complete.

Acknowledgements

The authors want to thank to José Miguel Cisternas, Eduardo Escobar, Hernán Fierro and Nicolás Ulriksen because of their contributions to this work. This research work has been partially supported by the Fondecyt Project (Chile), grant: 1120207, and also the LACCIR Project, grant: R1210LAC002.

References

[Antu12] Antunes, P., Herskovic, V., Ochoa, S.F., Pino, J.A.: Structuring Dimensions for Collaborative Systems Evaluation. ACM Computing Surveys, vol. 44(2), pp. 8:1-8:28 (2012)

[Been04] Beenen, G., Ling, K., Wang, X., Chang, K., Frankowski, D., Resnick, P., Kraut, R.E.: Using Social Psychology to Motivate Contributions to Online Communities. In: Proc. of CSCW '04. Chicago, United States (2004)

[Beng10] Benghazi, K., Noguera, M., Rodríguez-Domínguez, C., Pelegrina, A.B., Garrido, J.L.: Real-time Web Services Orchestration and Choreography. Proc. of the CAiSE 2010 Workshop EOMAS'10, Hammamet, Tunisia, pp. 142-153 (2010)

[Born11] Bornoe, N., Barkhuus, L.: Privacy Management in a Connected World: Students' Perception of Facebook Privacy Settings. In: Workshop on Collaborative Practices in Social Media CSCW '11. Hangzhou, China (2011)

[Busc96] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture: A System of Patterns. Wiley (1996)

[Chas80] Chase, I.D.: Social Process and Hierarchy Formation in Small Groups: A Comparative Perspective. American Sociological Review, vol. 45(6), pp. 905-924 (1980)

[Chen05] Cheng, R., Vassileva, J.: Adaptive Reward Mechanism for Sustainable Online Learning Community. Proc. of the 2005 Conference on Artificial Intelligence in Education (2005)

[Cran08] Crandall, D., Cosley, D., Huttenlocher, D., Kleinberg, J., Suri, S.: Feedback Effects between Similarity and Social Influence in Online Communities. In: Proc. of KDD '08. Las Vegas, United States (2008)

[Crum09] Crumlish, C., Malone, E.: Designing Social Interfaces. O'Reilly, Sebastopol (2009)

[Dell03] Dellarocas, C.: The Digitalization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. Management Science, vol. 49(10), pp. 1407-1424 (2003)

[DeSo03] De Souza, C.S., Preece, J.: A Framework for Analyzing and Understanding Online Communities. Interacting with Computers, vol. 16(3), pp. 579-610 (2004)

[Dour92] Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proc. of the 1992 ACM conference on Computer-supported cooperative work (1992)

[Dunb92] Dunbar R.I.M.: Neocortex Size as a Constraint on Group Size in Primates. Journal of Human Evolution, vol. 22(6), pp. 469-493 (1992)

[Dwye08] Dwyer, C., Hiltz, S.R., Widmeyer, G.: Understanding Development and Usage of Social Networking Services: The Social Software Performance Model. In: Proc. of the 41st Hawaii International Conference on System Sciences (HICSS '08). Hawaii, United States (2008)

[Elgg12] Elgg Foundation Project: Open Source Social Networking Engine. URL: http://www.elgg.org (Last visit: Jan. 10, 2012)

[Fern06] Fernando, O.N.N., Cohen, M.: Narrowcasting Attributes for Presence Awareness in Collaborative Virtual Environments. In: Proc. of the 6th IEEE International Conference on Computer and Information Technology (CIT '06). Seoul, South Korea (2006)

[Free00] Freedman, D.P., Weinberg, G.M.: Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products. Dorset House Publishing Co., Inc. (2000)

[Frie98] Friedkin, N.E.: A Structural Theory of Social Influence. Cambridge University Press (1998)

[Gart06] Garton, L., Haythornthwaite, C., Wellman, B.: Studying Online Social Networks. Journal of Computer-Mediated Communication, vol. 3(1) (2007)

[Girg02] Girgensohn, A., Lee, A.: Making Web Sites Be Places for Social Interaction. Proc. of the 2002 ACM Conference on CSCW, New Orleans, USA (2002)

[Gonc11] Gonçalves, B., Perra, N., Vespignani, A.: Modeling Users' Activity on Twitter Networks: Validation of Dunbar's Number. PLoS One, vol. 6(8) (2011)

[Guti11] Gutierrez, F., Baloian, N., Zurita, G.: Boosting Participation in Virtual Communities. Proc. of the CRIWG 2011. LNCS vol. 6969, pp. 14-29. Springer, Heidelberg (2011)

[Guti12] Gutierrez, F., Baloian, N., Ochoa, S.F., Zurita, G.: Designing the Software Support for Partially Virtual Communities. Proc. of the 19th Conference on Collaboration and Technology (CRIWG'12). LNCS vol. 7493, Springer-Verlag, pp. 73-88. Raesfeld, Germany (2012)

[Gutw98] Gutwin, C., Greenberg, S.: Effects of Awareness Support on Groupware Usability. In: Proc. of the ACM CHI '98 Human Factors in Computing Systems Conference. Los Angeles, CA, USA (1998)

[Hers11] Herskovic, V., Ochoa, S.F., Pino, J.A., Neyem, A.: The Iceberg Effect: Behind the User Interface of Mobile Collaborative Systems. Journal of Universal Computer Science, vol. 17, no. 2, pp. 183-202 (2011)

[Hers12] Herskovic, V., Neyem, A., Ochoa, S.F., Pino, J.A., Antunes, P.: Understanding Presence Awareness Information Needs Among Engineering Students. Proc. of the 16th IEEE Int. Conf. on Comp. Sup. Cooperative Work in Design (CSCWD´12), Wuhan, China (2012)

[Hill09] Hill, T., Supakkul, S., Chung, L.: Confirming and Reconfirming Architectural Decisions on Scalability: A Goal-Driven Simulation Approach. Proc. of the OTM 2009 Workshops. LNCS 5872, pp. 327-336. Springer, Heidelberg (2009)

[Howa10] Howard, T.: Design to Thrive: Creating Social Networks and Online Communities that Last. Morgan Kaufmann, San Francisco, USA (2010)

[Hugh89] Hughes, T.: The Evolution of Large Technological Systems. In: Bijker, W., Hughes, T., Pinch, T. (Eds.): The Social Construction of Technological Systems. The MIT Press: Cambridge, MA. (1989)

[Hunt09] Hunter, M.G., Stockdale, R.: Taxonomy of online communities: Ownership and value propositions. Proc. of the 42nd IEEE Hawaii Int. Conf. on System Sciences (2009)

[Irib09] Iriberri, A., Leroy, G.: A life-cycle perspective on online community success. ACM Computing Surveys 41(2) (2009)

[Kim00] Kim, A.J.: Community Building on the Web. Peachpit Press, Berkeley (2000)

[Klug96] Kluger, A.N., DeNisi, A.: The Effects of Feedback Interventions on Performance: A Historical Review, a Meta-Analysis, and a Preliminary Feedback Intervention Theory. Psychology Bulletin, vol. 119, pp. 254-384 (1996)

[Koll96] Kollock, P.: Design Principles for Online Communities. Proc. of the Harvard Conference on the Internet and Society (1996)

[Lamp07] Lampe, C.A.C., Ellison, N., Steinfield, C.: A Familiar Face(book): Profile Elements as Signal in an Online Social Network. In: Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems (2007)

[Lee01] Lee, A., Danis, C., Miller, T., Jung, Y.: Fostering Social Interaction in Online Spaces. Proc. of INTERACT ’01 (2001)

[Lee03] Lee, F.S., Vogel, D., Moez, L.: Virtual Community Informatics: A Review and Research Agenda. Journal of Information Technology Theory and Application, 5(1), 47-61 (2003)

[Liu11] Liu, Y., Gummadi, K.P., Krishnamurthy, B., Mislove, A.: Analyzing Facebook Privacy Settings: User Expectations vs. Reality. In: Proc. of the 2011 ACM SIGCOMM Conference on Internet Measurement. Berlin, Germany (2011)

[Lobr04] Lobry de Bruyn, L.: Monitoring Online Communication: Can the Development of Convergence and Social Presence Indicate an Interactive Learning Environment?. Distance Education, vol. 25(1) (2004)

[Marc06] Marcus, B., Machilek, F., Schütz, A.: Personality in Cyberspace: Personal Web Sites as Media for Personality Expressions and Impressions. Journal of Personality and Social Psychology, vol. 90(6), pp. 1014-1031 (2006)

[McMi86] McMillan, D.W., Chavis, D.M.: Sense of Community: A Definition and Theory. Journal of Community Psychology, vol. 14(1), pp. 6-23 (1986)

[McPh01] McPherson, L., Smith-Lovin, L., Cook, J.: Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology, vol. 27 (2001)

[Mena01] Menascé, D.A., Almeida, V.A.F.: Capacity Planning for Web Services: Metrics, Models and Methods. Prentice Hall, Upper Saddle River (2001)

[Misl10] Mislove, A., Viswanath, B., Gummadi, K.P., Druschel, P.: You Are Who You Know: Inferring User Profiles in Online Social Networks. Proc. of the WSDM'10, USA (2010)

[Mous09] Mousavidin, E. Goel, L.: A life cycle model of virtual communities. Proc. of the 42nd IEEE Hawaii International Conference on System Sciences (HICSS'09) (2009)

[Neye11] Neyem, A., Ochoa, S.F., Pino, J.A.: A Patterns System to Coordinate Mobile Groupware Applications. Group Decision and Negotiation vol. 20(5), pp. 563-592 (2011)

[Niel93] Nielsen, J.: Usability Engineering. Morgan Kaufmann: San Francisco, CA. (1993)

[Para07] Parameswaran, M.: Social Computing: An Overview. Communications of the Association for Information Systems (AIS) 19, 762-780 (2007)

[Plan04] Plant, R.: Online communities. Technology in Society 26, 51-65 (2004)

[Port04] Porter, C.E.: A Typology of Virtual Communities: A Multi-Disciplinary Foundation for Future Research. Journal of Computer-Mediated Communication vol. 10(1) (2004)

[Port08] Porter, J.: Designing for the Social Web. New Riders, Berkeley (2008)

[Pree01] Preece, J.: Sociability and Usability: Twenty Years of Chatting Online. Behavior and Information Technology Journal, vol. 20(5), pp. 347-356 (2001)

[Pree03] Preece, J., Maloney-Krichmar, D.: Online Communities. In: Jacko, J., Sears, A. (Eds.): Handbook of Human-Computer Interaction, Lawrence Erlbaum Associates Inc. Mahwah: New Jersey, pp. 596-620 (2003)

[Pree09] Preece, J., Shneiderman, B.: The reader-to-leader framework: motivating technology-mediated social participation. AIS Trans. on Human-Computer Interaction 1(1) (2009)

[Rams07] Ramsey, D., Beesley, K.B.: 'Perimeteritis' and rural health in Manitoba, Canada: Perspectives from rural healthcare managers. Rural and Remote Health 7, 850 (2007)

[Schu07] Schümmer, T., Lukosch, S.: Patterns for Computed-Mediated Interaction. John Wiley & Sons, Chichester (2007)

[Supa10] Supakkul, S., Hill, T., Chung, L., Tun, T.T., Leite, J.C.S.P.: An NFR Pattern Approach to Dealing with NFRs. Proc. of Requirements Engineering (RE'10), IEEE Press (2010)

[Tedj05] Tedjamulia, S.J., Olsen, D.R., Dean, D.L., Albrecht, C.C.: Motivating Content Contributions to Online Communities: Toward a More Comprehensive Theory. In: Proc. of the 38th Hawaii International Conference on System Sciences (HICSS). Hawaii, United States (2005)

[VanD06] Van Duyne, D.K., Landay, J.A., Hong, K.I.: The Design of Sites: Patterns for Creating Winning Web Sites. Prentice Hall, Upper Saddle River (2006)

[VanV99] Van Vugt, M., De Cremer, D.: Leadership in Social Dilemmas: Social Identification Effects on Collective Actions in Public Goods. Journal of Personality and Social Psychology, vol. 76(4), pp. 587-599 (1999)

[West09] Westerlund, M., Rajala, R., Nykänen, K., Järvensivu, T.: Trust and commitment in social networking – Lessons learned from two empirical studies. Proc. of the 25th IMP Conference, Marseille, France (2009)

[Xu05] Xu, L, Ziv, H., Richardson, D., Liu, Z.: Towards modeling non-functional requirements in software architecture. Proc. of Aspect-Oriented Software Design (AOSD'05); Workshop on Aspect-Oriented Requirements Engineering and Architecture Design (2005)