Designing the Software Support for Partially Virtual Communities

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

[1] Computer Science Department, Universidad de Chile
Av. Blanco Encalada 2120, 3rd Floor, Santiago, Chile
{frgutier,nbaloian,sochoa}@dcc.uchile.cl
[2] Control Management and Information Systems Department, Universidad de Chile
Diagonal Paraguay 257, Santiago, Chile
gzurita@fen.uchile.cl

Abstract. Designing software platforms to support the activities of partially virtual communities (PVC) is a challenging task since the supporting services must evolve continually according to the community evolution. Moreover, unsuitable supporting services usually lead the community to its demise. Therefore, these platforms must count on a flexible architecture that provides suitable services as a way to support interactions among community members, and thus contributing to keep the community sustainability. This article proposes a software architecture that helps software designers to address this challenge. Such a model can be used not only to ease the architectural design process, but also to evaluate already implemented PVC supporting systems. The article also shows a preliminary evaluation of both roles of the proposed model and discusses the obtained results.

Keywords: Social system architecture, software architecture, partially virtual communities, supporting systems.

1 Introduction

Over recent years, social computing has become present in many aspects of our daily activities. Although virtual communities have been present in several scenarios for some time, the recent rise of social computing systems has helped spread and diversify them. Several taxonomies have been proposed to classify these people associations [16, 28]. This article considers just one of these types that we have called *Partially Virtual Communities* (PVC) [11]. In such communities, members have the opportunity to interact frequently through both a virtual and a physical space. Examples of PVCs are the communities of a university course or people in a small neighborhood.

Membership in these communities is quite stable, meaning that few people join or quit these communities. PVCs depend on a certain personal interaction and knowledge among their members. Therefore, when two people decide to interact through the virtual space (i.e. the community supporting system), they know each other, and such contextual information (i.e. the mutual knowledge) allows them to appropriately interpret others' contributions. When a member makes a commitment, the rest of the participants know (or estimate) how trustworthy that commitment is,

V. Herskovic et al. (Eds.): CRIWG 2012, LNCS 7493, pp. 73–88, 2012.

based on the previous behavior of that person. The personal knowledge among members makes these communities stronger and tightly linked.

Since a PVC involves partially virtual participants, it inherits several features from physical and also from virtual communities. Although in a PVC its members cannot easily leave the community (because other links keep them connected; i.e. contractual links), several studies indicate that community members avoid participating when a certain number of conditions are not satisfied, finally leading to the community extinction [17, 25]. The unsuitability of services that support the interactions among community members usually acts as a trigger for such an end [14]. Therefore, ensuring the suitability of these services is mandatory for any platform that supports online and partially virtual communities. This service suitability is temporal, since it depends on the lifecycle stage living by the community. As long as the community evolves, some services become obsolete, requiring new ones. 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. Thus, designers can conceive a supporting architecture that is able to evolve with the community, and avoid the software support triggers the community demise.

To the best of our knowledge, the literature does not report 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.

This paper proposes a software architecture that helps design the architecture of PVC supporting systems. The model can be used as a design guideline for under development solutions, and also as an instrument to measure suitability of services embedded in already implemented platforms. The proposal was used to evaluate two already implemented PVC supporting systems, and also to design a particular PVC supporting platform. The obtained results are highly encouraging.

Next section defines the concept of a PVC. Section 3 presents the related work, which is focused on requirements and design guidelines for PVC supporting systems. Section 4 discusses a list of functional and non-functional requirements that should be considered when designing these systems. Section 5 proposes the software architecture for PVC supporting systems and it shows how its components help address the requirements presented in section 4. Section 6 presents a short-term field evaluation with expert and end users. Section 7 analyzes two commercial community-supporting systems and identifies their limitations to support interaction among members of a PVC. Finally, section 8 presents the conclusions and future work.

2 Partially Virtual Communities

A partially virtual community 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 physical community perform just face-to-face interactions, and members of a virtual community interact only through supporting systems (e.g. email or a Web application). Clearly, most communities involve physical and virtual interactions in several 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 close to a physical community, and a gamers community is a PVC that is probably close to a virtual community. In this article we consider the PVCs that are in the middle area of this spectrum (Fig. 1).

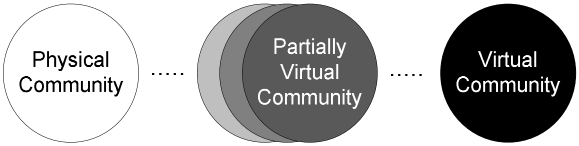

Fig. 1. Spectrum of communities according to the nature of their member interaction

There is a lack of consensus regarding an appropriate definition of the terms physical and virtual community [29]. Therefore, for physical communities we adhere to the definition given by Ramsey and Beesley, which indicates that they are a group of people who are bound together because of where they reside, work, visit or otherwise spend a continuous portion of their time [33]. 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"[21]. 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" [7], because physical and virtual communities seem to already accomplish with it [9]. This number indicates that human social networks involve stable relationships just in a range of between 100 and 200 individuals. These relationships are stable, when an individual knows who each person is, and how each person relates to every other.

Similar to physical and virtual communities, the PVC structure is diverse and it could be 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 rises spontaneously through member interaction is influenced by the organizational structure (in 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 [3]. In fact, whenever a group of people interacts within a community, a leader-follower relationship almost always emerges [38]. 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 [36].

3 Related Work

This section presents the main requirements to be considered by designers of software systems that support activities of a PVC. We then present and discuss the existing guidelines to model these supporting platforms.

3.1 Requirements for PVC Supporting Systems

PVC platforms typically support information dissemination, self-service transactions, communication and mediation [8]. 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 [4].

McMillan and Chavis [22] 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 lay within the basic requirements to be fulfilled by PVC platforms [39], whose design has to be driven by usability and sociability [31].

A persistent and updated identity triggers cooperation, as community members tend to identify each other and keep a track of their behavior in the past [19]. Moreover, user behavior and information published under personal profiles allows community members to infer relationships and/or other data related to different users [24]. 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* [20]. Information sharing, knowledge of group activity, and coordination are central to successful collaboration [6]. Collaborative systems like PVC platforms should consider context to support interaction among group members. In fact, users especially value information related to status and physical location, as well as profile information [13].

Although all these functional requirements (FR) identified in the literature are relevant in the design process, 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 [35, 40]. 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 communities studies, which should be considered when modeling these systems. For example, Preece and Shneiderman [32] 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.

Similarly, Kim [18] 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 [30] 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 [10]. 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 [15]. This model is based on four components: *remuneration*, *influence*, *belonging* and *significance*.

Concerning guidelines for social platforms, Crumlish [5] identifies a series of social interface design patterns and analyzes how they are applied into different systems. Van Duyne et al. [37] 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 that could be used as general guidelines for it. For example, Schümmer and Lukosch define a pattern language for computer-mediated interaction [34] 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.

4 Requirements to Support PVC Activities

This section identifies FR and NFR that are usually present in this type of supporting platforms. These requirements have been obtained from the literature review and from 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 [1].

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 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 [27].

When designing the interaction space, the supporting system should consider two disjointed environments: *public* and *private* [26]. Sharing resources between these two environments has to 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.

The platform architecture should also consider services that allow synchronous and asynchronous communication among community members [26]. 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. 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 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 [5, 15, 18, 30].
FR 02	The system should provide mechanisms for managing a personal identity by users [5, 8, 18, 30].
FR 03	The system should include awareness mechanisms in the form of users' availability, action identification and notifications [5, 6, 13].
FR 04	The system may include location awareness in order to allow face-to-face interactions and break the barriers linked to virtualness [5, 13].
FR 05	The system should allow and trigger relationship building among community members; e.g. friends, circles, groups [5, 8].
FR 06	The system may provide services for sharing content and media with other users, either in private groups or publicly [5, 10].
FR 07	The system should provide interaction mechanisms, such as synchronous and asynchronous communication modules [5, 18, 26, 37].
FR 08	The system may provide mechanisms for supporting collaboration and content creation among community members [5, 10].
FR 09	The system should include control mechanisms, such as peer moderation, governance structures and filters [5, 27, 37].
FR 10	The system should be designed following a motivation and participation strategy in order to ensure a certain level of activity through time [4, 8, 10, 32].
NFR 01	The system should react to short response times against any request made by users or its components [23].
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 [23].
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 [2].
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 [14].
NFR 05	The system should ensure privacy and security, as PVCs have to be trustworthy for users in order to trigger interactions [5, 11].
NFR 06	The system should be usable, since it has to support community member interaction and deal with different kinds of users [11, 31].

5 Software Architecture for PVC Supporting Systems

Herskovic et al. state that requirements of collaboration systems should be layered [12]. 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. 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.

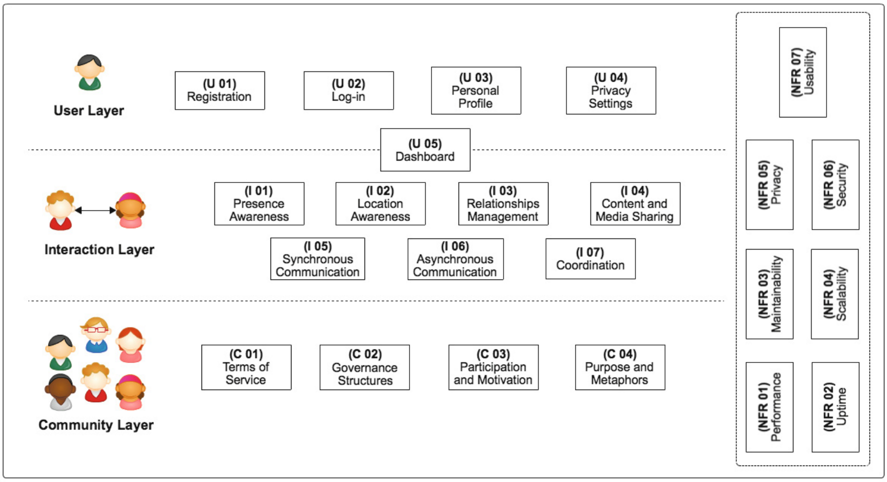

Fig. 2. Software architecture for PVC supporting systems

The User Layer is composed of five services; one of them is shared with the Interaction Layer. The *registration, log-in*, *personal profile*, and *privacy settings* manage the identity and visibility of a single community member. 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 Interaction Layer is composed of seven services: *presence awareness*, *location awareness*, *relationships management*, *content and media sharing*, *synchronous* and *asynchronous communication*, and *coordination*. 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 in 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: *terms of service*, *governance structures*, *participation and motivation strategies,* 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.

The complexity of 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 lives in the client side, and the two lower layers are in the server side. This design decision simplifies the services implementation.

Concerning the NFRs, they are "transversal requirements"; therefore, they affect 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 [2]. 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 [23]. 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 in the server side. The system *uptime* cannot be ensured through this architecture since it does not consider replicated components in the server side [23]; however it should be interesting to include 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. in the client and also in the server), it cannot be modified unless the user has a simultaneous access to both copies of such information. This information management policy is used also 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 architectures, this proposal addresses the systems *usability* just accomplishing with all previous requirements (including FR and NFR).

6 Using the Proposed Architecture

In order to determine if the services considered in the software architecture are suitable to support a real PVC, we have developed the supporting system for an existing community. This community was composed of 30 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 other students. The users participate through an avatar to keep anonymous their interactions. The community had a manager (an external person) who tracked the interactions and gave regularly feedback to members about their behavior in the platform. This community is still in service, but the tracking period was limited to 8 weeks from its initial launch. After that period we applied a survey to end-users to gather their feeling about the usefulness of the services provided by the platform. These services were completely aligned with the software 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 NFR considered. Using these results we tried to answer the following questions:

(Q1) Are the services considered in the architecture useful to support the interactions among PVC members?
(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 survey. Then we present the results obtained in the experimentation process with the users of the PVC platform (section 6.2), and also those gathered in the focus group with software designers (section 6.3).

6.1 Survey

The survey included an item for each service of the proposed architecture. Users rated the usefulness of such services using a 5-point Likert scale. Values of 1 and 2

Table 2. Description of supporting services

Service	Description
Personal Profile	Users have a personal space where they can manage their virtual identity. It provides support for an avatar, personal status or interests.
Privacy Settings	Users can decide what information will remain public and private. Also, they manage how they will receive notifications (e.g. email, in-site).
Dashboard	A main page where is published automatically the recent activity in the community, such as new messages and recent contributions.
Presence Awareness	Users can see the list of the other community members that are currently logged-in into the platform.
Location Awareness	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.
Relationships Management	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	The system supports media uploading (e.g. documents, pictures and videos), classifies it into categories and allows users to comment on them.
Synchronous Communication	The platform supports a video chat room for logged-in users. They have to allow camera and microphone access beforehand.
Asynchronous Communication	Users can publish, comment and rate discussions related to the different topics they have worked on the lecture sessions.
Coordination	The system provides a calendar with different permission levels: users can schedule activities that are private, or involve groups.

correspond to “negligible” services, a value of 3 corresponds to a “desirable” service, and a value of 4 or 5 means the service is “mandatory”. The survey also included an open comments section where users could suggest services to improve the system.

Some services considered in the model, such as registration and identification, were not considered in the survey, since they are either used only once, or required to access to the software support. Similarly, terms of service, 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.

6.2 Users Perception versus Designers Perception

Fig. 3 shows the usefulness of each service according to users and designers. Dark bars represent the average value assigned by the users to the usefulness of such services. Light ones show the numerical representation of the *usefulness perception level* according to the designers’ opinion. A continue scale from 0 to 10 was used to represent the usefulness of each service.

According to results shown in Fig. 3, 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.

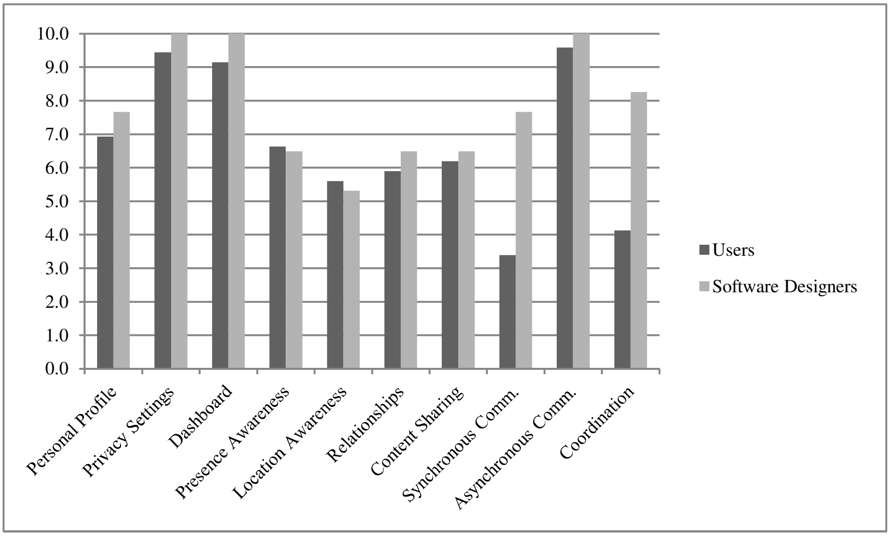

Fig. 3. Users vs. software designers’ usefulness perception of the services

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 agrees that a service that implements coordination mechanisms is desirable for this kind of community. However, end-users assigned a usefulness value considerably lower than expected one. This was also reflected on spontaneous comments that end-users stated at the end of the survey. The comments show a lack of initiative to use such service since it was not mandatory to perform the community activities during the experimentation period. The use of this service by either the community manager or other users would have motivated that community members use it consequently. Figure 4 shows the declared and the perceived usefulness assigned by end-users. Dark bars are the result of using the current implementation of a service. Light bars represent the value of each service (according to users' opinion) when they are properly implemented.

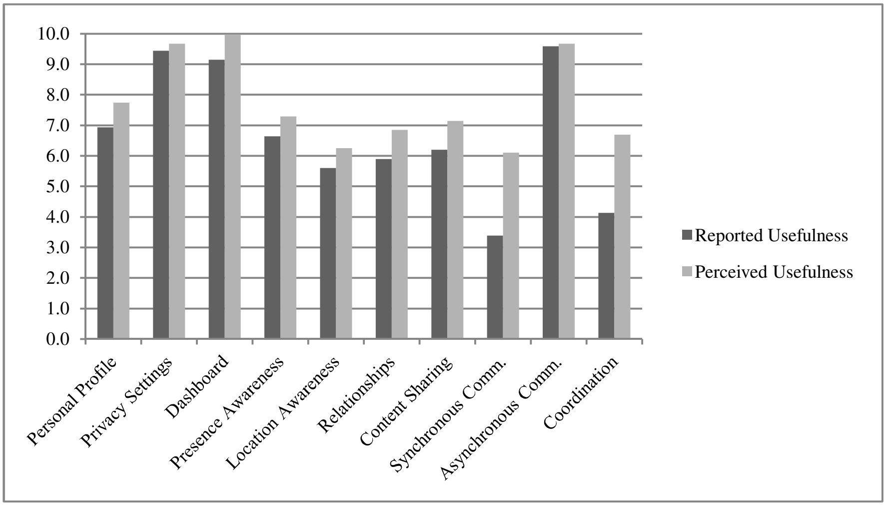

Fig. 4. Reported vs. perceived usefulness according to users

These results show that all services considered in the software architecture are considered useful by 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 for using the service (that is the case of the coordination service).

6.3 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 pointed out that this model is almost complete, as it lacks just of support for activity awareness. This provides a preliminary response to Q2, which is also supported by the results shown in Fig. 3. All designers considered this architecture to be useful for analyzing other services in different contexts, and not difficult 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.

7 Analysis of Already Implemented PVC Supporting Systems

In order to show how the proposed architecture can be applied in practice, we will briefly analyze two commercial PVC supporting systems: *Facebook* and *U-Cursos*. In this analysis we will try attempt to verify whether or not these systems satisfy the set of requirements specified in section 3, and also if the non-addressed requirements are required by the community members. Thus, we intend to show that this proposal can be used for: (1) designing new systems, (2) choose an already implemented system from a set of possibilities, and (3) identify further customizations or extensions needed to include in a supporting system that is currently used by a specific community.

7.1 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. Fig. 5 shows a typical page for a user and it identifies the main components that match with the proposed architecture.

We can see most services considered in the architecture are part of Facebook. However, two services usually required by PVC were not included: location awareness (i.e. positioning of community members) and coordination mechanisms (e.g. community agenda or community members commitments). Only location awareness is partially supported by the use of geo-tagging, and there are no simple mechanisms for coordinating community members and activities. This result is not surprising because Facebook was not particularly designed to support PVCs. However, this fact allows us to show that the proposed architecture can be used as a reference to identify mandatory services in PVC supporting systems.

7.2 U-Cursos

U-Cursos 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. Fig. 6 shows the main user interface of *U-Cursos*.

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

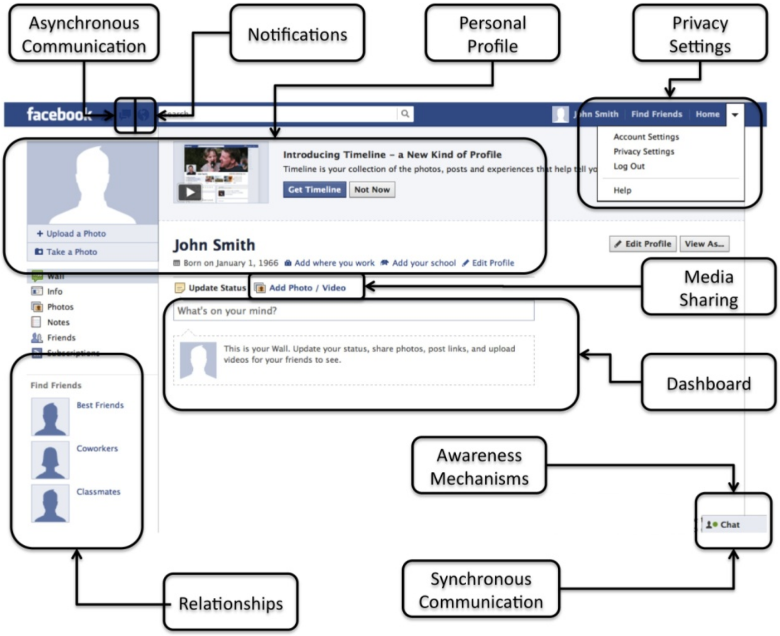

Fig. 5. Facebook page for a user

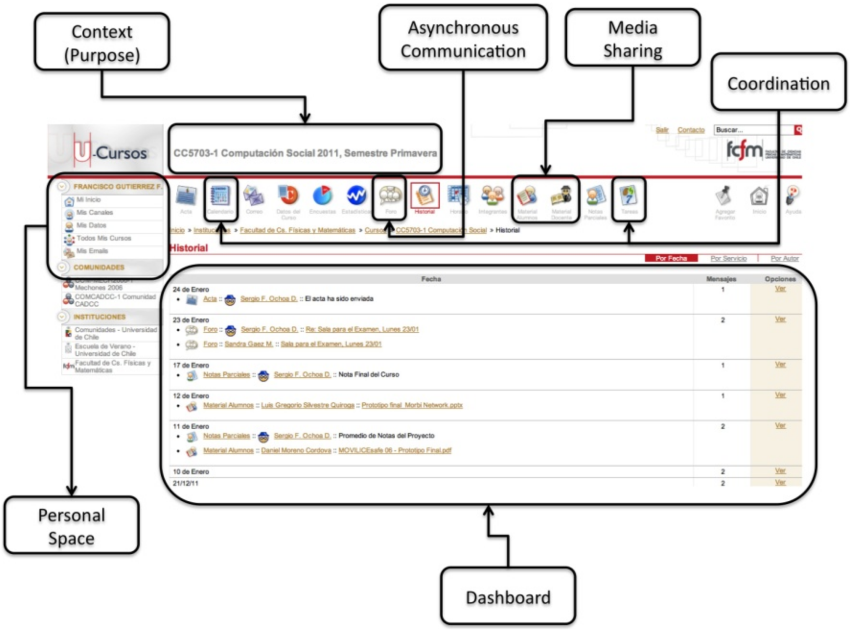

Fig. 6. U-Cursos main user interface

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 indentified in an early stage of the system development.

8 Conclusions and Future Work

A PVC is a hybrid between a physical and a virtual community, and we define a PVC *a group of people who interact around a shared interest or goal using technology-mediated and face-to-face mechanisms*.

This article identifies a list of recurrent requirements that should be considered when designing the architecture of a PVC supporting system. These requirements come from a literature review and also from the authors' experience developing and evaluating PVC supporting systems. Based on those requirements, a preliminary architectural model was proposed as a reference for under development and already implemented PVC supporting systems. The architecture is layered which contributes to the software maintainability and extensibility. This capability is important in these systems because they are in constant evolution. A number of recurrent services were identified as part of the architecture layers.

In order to perform a preliminary validation of the proposal, we carried out a short-term study with participants in an undergraduate course at the University of Chile. We have also conducted a focus group with expert users to examine the real and the expected usefulness of the services considered in the architecture. Such activity allows us to envision that the proposal could be used for evaluating the design of already implemented PVC software platforms.

Three research questions were stated in this article. Answering these questions will require evaluating the proposal more in-depth. However, the interim results provide a first response, which indicates the proposed software architecture considers services useful to support interaction among members of a PVC (*Q1*). This would also be useful to support the design of these systems (*Q2*) and the evaluation already implemented platforms (*Q3*).

Trying to answer the research questions 2 and 3 we have analyzed two commercial PVC supporting systems. The analysis was done through a focus group with software designers. The preliminary results indicate the proposed architecture can be used to (1) help software designers to model new PVC supporting systems, (2) identify suitable alternatives from a set of already implemented systems, and (3) determine mandatory services to be included in systems that are into production. The next steps in this work consider to conduct a survey to members of various PVCs in order to determine if the services identified in the software architecture are complete and also if all of them are mandatory.

Acknowledgements. This work has been partially supported by the Fondecyt Project (Chile), grant: 1120207, and also the LACCIR project, grant: R1210LAC002.

References

1. Benghazi, K., Noguera, M., Rodríguez-Domínguez, C., Pelegrina, A.B., Garrido, J.L.: Real-time Web Services Orchestration and Choreography. In: Proc. of the CAiSE 2010 Workshop EOMAS 2010, Hammamet, Tunisia, pp. 142–153 (2010)
2. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-Oriented Software Architecture: A System of Patterns. Wiley (1996)
3. Chase, I.D.: Social Process and Hierarchy Formation in Small Groups: A Comparative Perspective. American Sociological Review 45(6), 905–924 (1980)
4. Cheng, R., Vassileva, J.: Adaptive Reward Mechanism for Sustainable Online Learning Community. In: Proc. of the 2005 Conference on Artificial Intelligence in Education (2005)
5. Crumlish, C., Malone, E.: Designing Social Interfaces. O'Reilly, Sebastopol (2009)
6. Dourish, P., Bellotti, V.: Awareness and Coordination in Shared Workspaces. In: Proc. of the 1992 ACM Conference on Computer-Supported Cooperative Work (1992)
7. Dunbar, R.I.M.: Neocortex Size as a Constraint on Group Size in Primates. Journal of Human Evolution 22(6), 469–493 (1992)
8. Girgensohn, A., Lee, A.: Making Web Sites Be Places for Social Interaction. In: Proc. of the 2002 ACM Conference on CSCW. ACM Press, New Orleans (2002)
9. Gonçalves, B., Perra, N., Vespignani, A.: Modeling Users' Activity on Twitter Networks: Validation of Dunbar's Number. PLoS One 6(8) (2011)
10. Gutierrez, F., Baloian, N., Zurita, G.: Boosting Participation in Virtual Communities. In: Vivacqua, A.S., Gutwin, C., Borges, M.R.S. (eds.) CRIWG 2011. LNCS, vol. 6969, pp. 14–29. Springer, Heidelberg (2011)
11. Gutierrez, F., Baloian, N., Ochoa, S.F., Zurita, G.: A Conceptual Model to Design Partially Virtual Communities. In: Proc. of the 16th IEEE Int. Conf. on Computer Supported Cooperative Work in Design (CSCWD 2012), Wuhan, China, May 23 -25 (2012)
12. 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 17(2), 183–202 (2011)
13. Herskovic, V., Neyem, A., Ochoa, S.F., Pino, J.A., Antunes, P.: Understanding Presence Awareness Information Needs Among Engineering Students. In: Proc. of the 16th IEEE Int. Conf. on Comp. Sup. Cooperative Work in Design (CSCWD 2012), China, May 23-25 (2012)
14. Hill, T., Supakkul, S., Chung, L.: Confirming and Reconfirming Architectural Decisions on Scalability: A Goal-Driven Simulation Approach. In: Meersman, R., Herrero, P., Dillon, T. (eds.) OTM 2009 Workshops. LNCS, vol. 5872, pp. 327–336. Springer, Heidelberg (2009)
15. Howard, T.: Design to Thrive: Creating Social Networks and Online Communities that Last. Morgan Kaufmann, San Francisco (2010)
16. Hunter, M.G., Stockdale, R.: Taxonomy of online communities: Ownership and value propositions. In: Proc. of the 42nd IEEE Hawaii Int. Conf. on System Sciences (2009)
17. Iriberri, A., Leroy, G.: A life-cycle perspective on online community success. ACM Computing Surveys 41(2) (2009)
18. Kim, A.J.: Community Building on the Web. Peachpit Press, Berkeley (2000)

19. Kollock, P.: Design Principles for Online Communities. In: Proc. of the Harvard Conference on the Internet and Society (1996)
20. Lee, A., Danis, C., Miller, T., Jung, Y.: Fostering Social Interaction in Online Spaces. In: Proc. of INTERACT 2001 (2001)
21. 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)
22. McMillan, D.W., Chavis, D.M.: Sense of Community: A Definition and Theory. Journal of Community Psychology 14(1), 6–23 (1986)
23. Menascé, D.A., Almeida, V.A.F.: Capacity Planning for Web Services: Metrics, Models and Methods. Prentice Hall, Upper Saddle River (2001)
24. Mislove, A., Viswanath, B., Gummadi, K.P., Druschel, P.: You Are Who You Know: Inferring User Profiles in Online Social Networks. In: Proc. of the WSDM 2010, USA (2010)
25. Mousavidin, E., Goel, L.: A life cycle model of virtual communities. In: Proc. of the 42nd IEEE Hawaii International Conference on System Sciences, HICSS 2009 (2009)
26. Neyem, A., Ochoa, S.F., Pino, J.A.: A Patterns System to Coordinate Mobile Groupware Applications. Group Decision and Negotiation 20(5), 563–592 (2011)
27. Parameswaran, M.: Social Computing: An Overview. Communications of the Association for Information Systems (AIS) 19, 762–780 (2007)
28. Plant, R.: Online communities. Technology in Society 26, 51–65 (2004)
29. Porter, C.E.: A Typology of Virtual Communities: A Multi-Disciplinary Foundation for Future Research. Journal of Computer-Mediated Communication 10(1) (2004)
30. Porter, J.: Designing for the Social Web. New Riders, Berkeley (2008)
31. Preece, J.: Sociability and Usability in Online Communities: Determining and Measuring Success. Behavior and Information Technology Journal 20(5), 347–356 (2001)
32. Preece, J., Shneiderman, B.: The reader-to-leader framework: motivating technology-mediated social participation. AIS Trans. on Human-Computer Interaction 1(1) (2009)
33. Ramsey, D., Beesley, K.B.: 'Perimeteritis' and rural health in Manitoba, Canada: Perspectives from rural healthcare managers. Rural and Remote Health 7, 850 (2007)
34. Schümmer, T., Lukosch, S.: Patterns for Computed-Mediated Interaction. John Wiley & Sons, Chichester (2007)
35. Supakkul, S., Hill, T., Chung, L., Tun, T.T., Leite, J.C.S.P.: An NFR Pattern Approach to Dealing with NFRs. In: Proc. of Requirements Engineering (RE 2010). IEEE Press (2010)
36. Tedjamulia, S., Olsen, D., Dean, D., Albrecht, C.: Motivating Content Contributions to Online Communities: Toward a More Comprehensive Theory. In: Proc. of IEEE Hawaii International Conference on System Sciences (2005)
37. 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)
38. 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 76(4), 587–599 (1999)
39. Westerlund, M., Rajala, R., Nykänen, K., Järvensivu, T.: Trust and commitment in social networking - Lessons learned from two empirical studies. In: Proc. of the 25th IMP Conference, Marseille, France (2009)
40. Xu, L., Ziv, H., Richardson, D., Liu, Z.: Towards modeling non-functional requirements in software architecture. In: Proc. of Aspect-Oriented Software Design (AOSD 2005); Workshop on Aspect-Oriented Requirements Engineering and Architecture Design (2005)