Computing and Informatics, Vol. 34, 2015, 521–558

AN ARCHITECTURE TO SUPPORT THE DESIGN AND EVALUATION OF SOFTWARE PLATFORMS FOR PARTIALLY VIRTUAL COMMUNITIES

Francisco J. Gutierrez, Sergio F. Ochoa, Nelson Baloian

Computer Science Department
University of Chile
Beauchef 851, 3rd floor
Santiago, Chile
e-mail: {frgutier, sochoa, nbaloian}@dcc.uchile.cl

Gustavo Zurita

Management and Information Systems Department
University of Chile
Diagonal Paraguay 257, 13th floor
Santiago, Chile
e-mail: gzurita@fen.uchile.cl

Luis Loyola

R & D Department
SkillUpJapan Corporation
Tokyo, Japan
e-mail: loyola@skillupjapan.co.jp

Abstract. Several researchers have identified the design of collaborative systems as a particularly challenging task, because it has to consider services that are not easily identifiable by software designers. The systems that support the activities of partially virtual communities (PVC) are not an exception. Typically, designers of PVC supporting systems are able to identify services that have a visual representation on the user interface, but they usually overlook those that run in the system back-end. In an attempt to help designers to consider these two types of services, this article proposes a software architecture that can be used for both, designing new PVC supporting systems and evaluating 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 [30, 49]. One particular type of these associations is what we call *Partially Virtual Communities* (PVC) [23], where the community members have the opportunity to interact frequently through both a virtual and a physical space. Members typically know each other, and such contextual information allows them to appropriately interpret the contributions made by others. 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 and eventually stop using the community supporting systems when a number of conditions are not met. When this behavior becomes recurrent, the community typically is pushed toward its demise [31, 45, 27].

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 life cycle. To the best of our knowledge, the literature does not report works presenting structural designs that help address this challenge; therefore designers of this type of system must improvise or adopt ad hoc solutions to deal with this issue. Moreover, software developers typically have to face the *iceberg effect*, which is usually present in the development of collaborative systems [25]. Using the metaphor of an iceberg, Herskovic et al. [25] 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 in the user interface makes these services not easy to identify. Examples of it are the services required to support the community governance and monitoring, or the tracking of members' activities. If designers have no previous experience modeling this type of system, it is highly probable that several of these services will not be considered in the design of the PVC supporting system. A deficient design will lead the system towards a limited implementation, which will negatively affect the support offered to the community activities.

In this article we propose a software architecture that helps design PVC supporting platforms and evaluate already implemented systems. Particularly, this architecture provides general design guidelines and feedback to developers about services to be considered in the new supporting platform. Moreover, the evaluation of already implemented systems can be done, by comparing the services provided by the implemented system and those considered in the proposed architecture. The pertinence of these latter 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 within 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 cases where the proposed architecture was used as a design guideline during the development of particular PVC supporting systems. Section 8 reports three cases 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 as those communities that are in the middle range of this spectrum (Figure 1).

There is a lack of consensus regarding an appropriate definition of the terms physical and virtual community [50]. Therefore, for physical communities we adhere to the definition given by Ramsey and Beesley, which indicates that they correspond

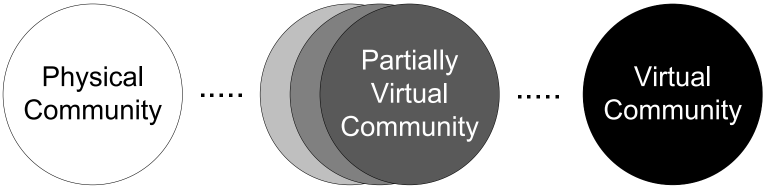


Figure 1. Spectrum of communities according to the nature of their member interaction

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 [55]. 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" [37]. 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 [13], 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. Moreover, according to Gonçalves et al. [21], this number also appears to characterize stable relationships in virtual communities.

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. Given that physical communities tend to assume a hierarchical structure [6], we can infer that this kind of organization will also be present in PVC. In fact, these social constructs can be understood as physical groups augmented with computer-mediated communication mechanisms, or as virtual groups where a face-to-face interaction scenario tends to generate a sense of community. Moreover, whenever a group of people interacts within a physical community, a leader-follower relationship almost always emerges [61]. This pattern has also been observed in the natural evolution in the life cycle of online communities [54, 57]. 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 [59]. Finally, Walther [62] describes that computer-mediated communication can actually reflect personal bonds, being sometimes "as personal as face-to-face" interaction, or even surpassing these bonds in some interpersonal aspects. In fact, the author claims that the combination of media attributes, social phenomena, and other socio-psychological processes, may lead computer-mediated communication to become hyperpersonal (i.e. exceed face-to-face interpersonal communication), reflecting thus how interaction mechanisms can be possible in PVCs.

3 RELATED WORK

This section presents an overview of the main requirements reported in literature for systems supporting the activities in a PVC. It also presents and discusses existing guidelines to model 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 [20]. 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 [7].

McMillan and Chavis [41] state there are four elements that define a sense of community: membership, influence, integration and fulfillment of needs, and a shared emotional connection. Since PVC supporting systems aim to facilitate computer-mediated and face-to-face interaction, encouraging online participation and allowing social interaction represents basic requirements to be fulfilled by these platforms [63].

People in a community tend to be similar to those whom they have close connections with. This is explained by two reasons: (1) people make communities grow to resemble their current friends due to social influence [18], and (2) people tend to form new links to others who are already like them [42]. While both factors are present in everyday social processes, according to Crandall et al. [8] they are in tension; e.g. social influence can push systems toward behavior uniformity, while selection can lead to fragmentation. In fact, social interaction is both an effect and a cause of selection, as there are clear feedback effects between these two factors. 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 [11]. Therefore, these mechanisms may affect the future behavior of the entire community towards that user.

A persistent and updated identity triggers cooperation, because community members tend to identify each other and keep a track of their past behavior [34]. Moreover, the behavior and the information published by community members under their personal profiles allow other members to infer relationships and features of the participants [44]. 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* [36]. Also, information sharing, the knowledge of group activity and coordination are central to successful collaboration [12]. 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 [26].

Although all these functional requirements (FR) identified in 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 to 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 the architectural design [58, 64]. Such 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 system users, particularly in PVCs where its members' interaction is 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 [54] 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 [29] claims that large technological systems are complex and messy with no clear boundaries. Therefore, people within a technological system have a critical role, that is to complete a feedback loop, by perceiving the gap between the system performance and the system goals. In fact, it is only through this feedback loop that errors are caught and corrected, leading to an improvement in system performance. Kluger and DeNisi [33] define feedback as the actions taken by external agents to provide information regarding some aspects of one's task. Tedjamulia et al. [59] propose that performance feedback and social recognition are often used in online communities because they allow self-organization of contributions in large sites; and they bring to bear the collective, distributed, and significant human resources available in a community.

Beenen et al. [2] 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. Similarly, Kim [32] 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 [51] 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. [22] state that participation is a key metric for evaluating the success of an online community. 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 [28] proposes a model to address the community member behavior and tries to identify the services required by them. This model is based on four components: *remuneration*, *influence*, *belonging* and *significance*.

Concerning guidelines for social platforms, Crumlish [9] identifies a series of social interface design patterns and analyzes how they are applied into different systems. Van Duyne et al. [60] 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 do not explicitly support the physical interactions that are required by PVC members.

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 [56] define a pattern language for computer-mediated interaction 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. [1] propose a framework to evaluate this kind of systems. 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* [10, 52, 53]. In the field of Human-Computer Interaction, Nielsen [47] 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*) [10, 52, 53]. Among these factors, we can state governance structures such as moderation mechanisms, to the extent of establishing a common cultural context in the community.

In the present article, we aim to extend the current research literature in online communities by presenting a software architecture to support the design and evaluation of PVC supporting systems. In fact, PVCs include characteristics from both, physical communities (as how they are understood in psychology and sociology), and online communities (as how they are understood in human-computer interaction and computer-mediated communication). In our proposal we include several recurrent services that have been reported in the literature of online communities, and also services identified by us during the development and evaluation of PVC supporting systems with a particular interest in face-to-face interaction, awareness and coordination mechanisms.

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 platform. These requirements have been obtained from the literature review and the authors' past experiences as designers of these software platforms.

Typically, PVC supporting systems 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 [3].

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 [48, 53].

In online communities, users generally need to express and expose their identities in a social context [40]. 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 [44]. However, the disclosure of the online identity of a user may present a certain number of issues related to privacy [4, 38] 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 [22]. 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 [19, 35, 39].

When designing the interaction space, the supporting system should consider two disjointed environments: *public* and *private* [46]. 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 [14, 32].

The platform architecture should also consider services that allow synchronous and asynchronous communication, as well as coordination mechanisms among community members [46]. The architecture 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 [24]. Since the community is partially physical, user location awareness mechanisms should be considered when aiming to facilitate the face-to-face interaction.

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.

5 SOFTWARE ARCHITECTURE FOR PVC SYSTEMS

Herskovic et al. [26] 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 users through the application front-end. Following this line of reasoning, we propose a software architecture composed of three layers (Figure 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.

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.

Req.	**Description**
FR 01	The system should provide registration mechanisms that facilitate the appropriation of the platform by its users [9, 28, 32, 40, 51].
FR 02	The system should provide mechanisms for managing a personal identity by its users [9, 20, 32, 51].
FR 03	The system should provide awareness in the form of users' availability, action identification and notifications [9, 12, 16, 26].
FR 04	The system may include location awareness to allow face-to-face interaction and break the barriers linked to virtualness [9, 26].
FR 05	The system should allow and trigger relationship building among community members; e.g. friends, circles, groups [9, 20].
FR 06	The system may provide services for sharing content and media with other users, either in private groups or publicly [9, 22].
FR 07	The system should provide interaction mechanisms, like synchronous and asynchronous communication modules [9, 32, 46, 60].
FR 08	The system should provide appropriate feedback to its users on the different actions performed by them [14, 32].
FR 09	The system may provide coordination support, such as creating content collaboratively, among community members [9, 22].
FR 10	The system should include control mechanisms, such as peer moderation, governance structures and filters [9, 48, 53, 60].
FR 11	The system should follow a motivation and participation strategy to ensure a certain level of activity through time [7, 20, 22, 54].
FR 12	The system should allow monitoring user activity and contributions performed in the software support through time [19, 35, 39].
NFR 01	The system should quickly react to any request made by users or its components [43].
NFR 02	The system should be highly available, since PVCs are supposed to allow members interact at any time [43].
NFR 03	The system should be extensible, because communities naturally evolve in time and follow a lifecycle, as well as its users [5].
NFR 04	The system should be scalable, in order to handle a growing number of users and contributions made within the community [27].
NFR 05	The system should ensure privacy and security, as PVCs have to be trustworthy for users in order to trigger interactions [4, 9, 22, 38].
NFR 06	The system should be usable, since it has to support member interaction and deal with different kinds of users [22, 52].

Table 1. Requirements for a PVC Supporting System

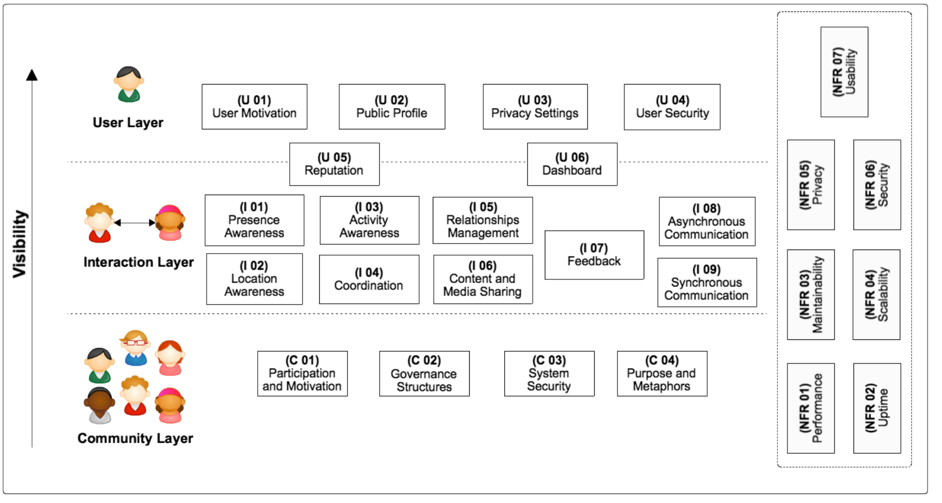


Figure 2. Software architecture for PVC supporting systems

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 the face-to-face interaction, 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 the 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 systems using layers [5]. We can also expect an appropriate *performance* of the systems that are implemented using this architecture because it follows the client-server architectural style and involves just three layers [43]. 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 the server side. The system *uptime* cannot be ensured through this architecture since it does not consider replicated components in the server side [43]. It should be interesting to include this requirement 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 Figure 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 Figure 2. Since NFR are transversal to the three layers, they are not depicted in the table.

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. [26], 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. Regarding 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

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**
U1	X	X										
U2		X										
U3		X										X
U4		X										X
U5		X						X		X	X	
U6		X						X			X	
I1			X									
I2				X								
I3			X									
I4									X			
I5					X							
I6						X						
I7								X				
I8							X					
I9							X					
C1											X	
C2										X		
C3												X
C4											X	

Table 2. Requirements traceability matrix

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 evaluation instrument respectively.

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. This can be achieved by stating the list of the stakeholders' concerns that have to be covered by the supporting system, relating these concerns to building blocks provided in Figure 2 (i.e. stating which are the specific requirements that need to be satisfied in terms of user services, interaction services, and community services), and finally, translating these concerns into particular software requirements (functional and non-functional) that are going to be used by designers for describing the particular software architecture of the envisaged PVC supporting

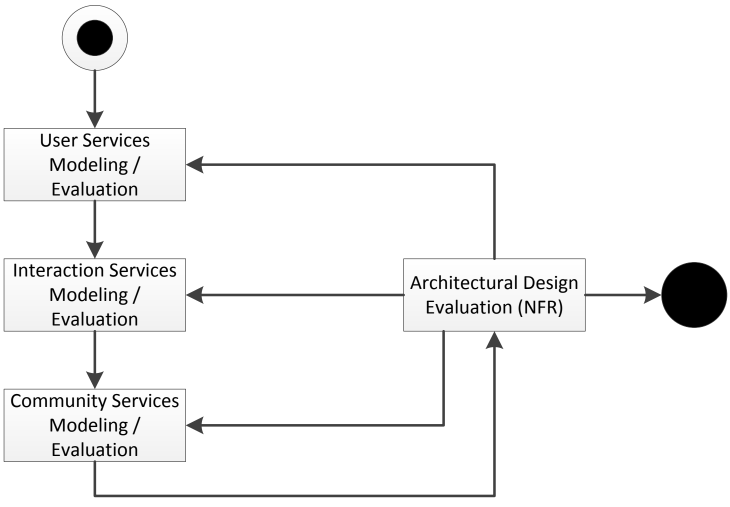


Figure 3. Process to use the proposed architecture

system. Therefore, it would be there possible to enhance non-functional qualities of the implemented solution, such as maintainability, extensibility, performance, scalability and uptime. Once this basic structure for the new system has been established, 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 Figure 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 fill a traceability matrix that matches the functional (FR) and non-functional requirements (NFR) of the selected services. Thus, 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, recommendable 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 make one or more decisions regarding them.

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 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 a formal technical review [17]. 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 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 or cannot have access to 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 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 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 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, having 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. Since 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

This section shows how the proposed architecture can be used as a guideline for designing PVC supporting systems. Three case studies are presented and discussed. The first two cases involved advanced undergraduate students, whom computer science in the 10^{th} or 11^{th} term at the University of Chile was major. The students worked in pairs for 16 weeks to develop a detailed design of a particular PVC supporting system for an existent community. During such a time period, the students have to conceive, design and preliminarily validate the system proposal. This design process involved several checkpoints in which software inspectors used the proposed architecture as the instrument to identify mandatory and optional supporting services. Such experience allowed us to obtain a preliminary validation about the suitability of the design guidelines considered in the proposed architecture.

The third case study was the development of a platform that supports activities of a community formed by the students, teaching assistants and instructors of a course at the University of Chile Business School. An experienced software engineer participated in the development of the PVC supporting platform, who used the proposed architecture as a guideline for the product design.

In the three case studies the authors monitored the development process, the product evolution and the usefulness of the proposed architecture. The main goal was to get preliminary answers to the following research questions:

(RQ1): How suitable are the services considered in the architecture to support the interactions among members of a PVC?

(RQ2): How well does the architecture support the evaluation of these system designs?

The answers to these questions show us the completeness and usefulness of this proposal. The following subsections describe the case studies, the evaluation methods used in these cases and the obtained results.

7.1 Case Studies

The products to be developed in the first two case studies were similar in terms of the development effort, however the third one was larger and also more complex than the two previous projects. The list of FR and NFR to be addressed were different in each case. Moreover, no major services were identified as mandatory for these projects, except those already considered in the architecture. This conveys the idea that the proposal would provide an important coverage of services for these applications. Next, we briefly describe the products to be developed in each case study.

7.1.1 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 proper mechanisms for managing their community members, supporting their interaction, 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 interaction among community members around these events.

The proposed solution was designed thinking on 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.

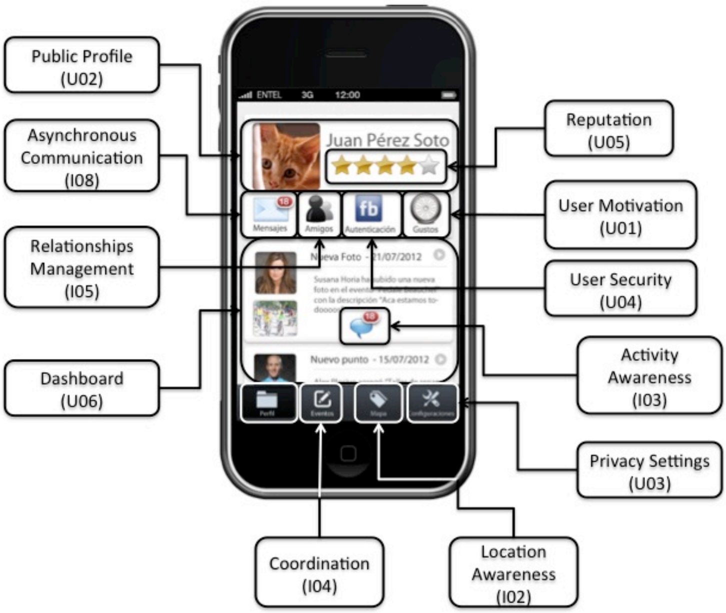


Figure 4. Main user interface of the bikers' community mobile application

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.

7.1.2 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 of a laptop/desktop as the target device. Figure 5 shows the main user interface of the proposed solution.

Similar to the previous case, the users have a personal profile, a reputation and a role. The roles go from regular user to community manager. The user role evolves depending on the user's actions and the recognition that the community gives to that member. The role determines the type of actions that can be performed by a user.

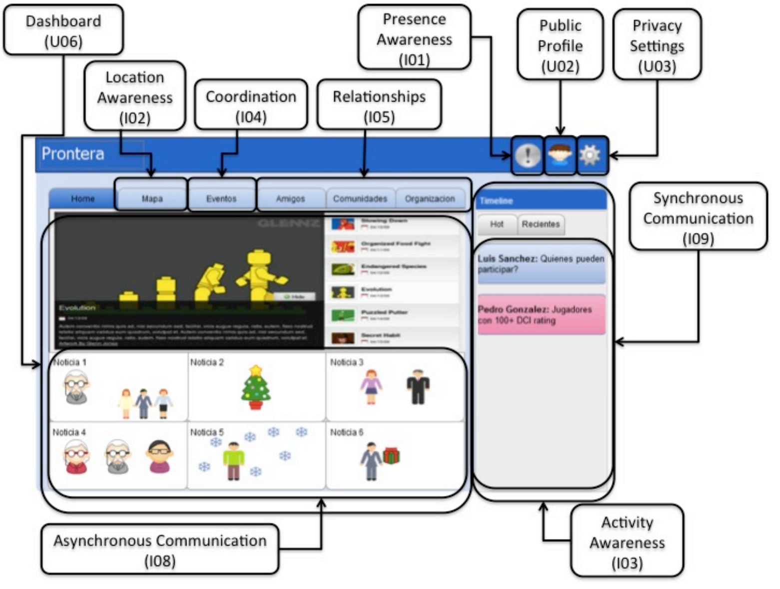


Figure 5. Main user interface of the players' community

7.1.3 Supporting a Course Community

The software developed in this case study supported the activities of thirty students enrolled in an introductory *Information Technology* course of the Business School at the University of Chile. Students taking part in this experience were volunteers and 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 the community members and participated in the course discussions. The users utilized an avatar and a pseudonym to maintain anonymity during their interaction. The community had a manager (an external user) who tracked the interaction and gave a regular feedback to members about their participation in this virtual scenario. The tracking period was limited to 8 weeks from its initial launch.

This platform was implemented as a desktop web application, by modifying and extending the core functionalities offered by the PHP framework Elgg [15]. 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 prototype allows us to validate 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.

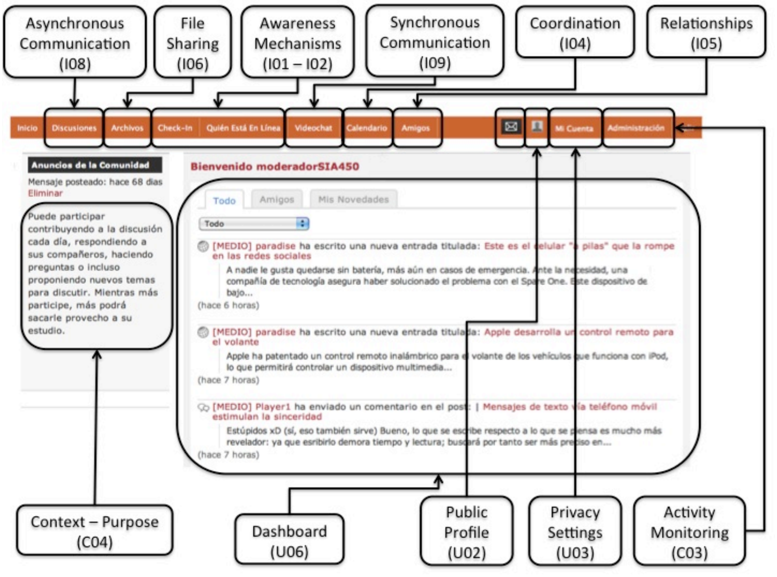


Figure 6. Main user interface of the developed platform

7.2 Evaluation Methods

The evaluation methods considered in the first two case studies were the same, as the projects were developed as a part of the same undergraduate course during the same term. The third case study involved particular evaluation methods, because in that case the proposed architecture was used in a way different from the previous cases. Next we describe the instruments used in these case studies.

7.2.1 Evaluation Methods for the Bikers and Players Communities

Due to the fact that the design of these applications was part of the student work in a formal Computer Science course, it involved an incremental process with five stages:

1. to research, conceptualize and characterize the communities to be supported by each application, and also the problem to be addressed in these projects;
2. to establish the user requirements to be considered and perform a preliminary design of each platform;
3. to perform an architectural design of these platforms;
4. to build a detailed design of each supporting system; and
5. to present a functional prototype of the designed platforms.

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 also a Formal Technical Review (FTR) on such a product.

The FTR followed the dynamic established for the *inspection* [17], which is regularly used in software projects to determine the status and quality of a product. Each inspection involved 50 minutes approximately per project: 15–20 minutes for presenting the product design and 30–35 minutes for answering the inspectors' questions. The inspectors participating in the FTR (i.e. the instructor and the teaching assistant) were experienced in both the use of inspection activities to verify the status and quality of software products, and the development of social computing platforms. These inspectors used the proposed architecture as a guideline to identify the services that could be useful to support the activities of the community members.

During these sessions the inspectors can only ask questions and make comments, but no suggestions. Depending on the inspection results, the designers can eventually make new design decisions and changes in the product design. Although the students (i.e. designers) knew about the services typically required by these supporting systems, they did not count on an instrument (e.g. the architecture) to validate their designs.

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 had to 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 was an already implemented product. Using such a prototype, the instructor, the teaching assistant and the rest of the students evaluated the functionality of the platform. Moreover, the inspectors evaluated the relevance that the services proposed in the architecture have to the design of these products. A formal grade was assigned to each prototype according to the usability and usefulness perceived by the inspectors.

7.2.2 Evaluation Methods for the Course Community

The system supporting the course community activities was conceived based on the proposed architecture, and the developer that implemented the system was also involved in the architecture definition. For that reason we did not consider this system developer in the evaluation process.

In order to identify the value that the services included in the architecture have for the users, we evaluated the system over a period of eight weeks with 23 members of the course community. After that period we conducted a survey on the participants to gather their opinions about the usefulness of the system services. The survey considered only the services from the architecture that are visible for the users, and it was conducted during a lecture of the course. The participants were able to ask the lecturer for additional information in case of need. Users graded the usefulness of those services using a 5-point Likert scale. A value of 1 corresponds to a "negligible" service, 2 corresponds to a service that is "dispensable", 3 corresponds to a "desirable" service, 4 means "recommendable" and 5 indicates that the service is "mandatory". The survey also included a section for open comments where the users could suggest services that help 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 the users. Table 3 summarizes the services considered in the survey.

Additionally, we carried out a focus group with six software engineers to analyze the suitability of services provided by the system, and the pertinence of the NFR considered in the system design. Before and during the focus group these engineers had access not only to the system prototype, but also to its design. Two participants had experience in the design of social platforms, two had experience evaluating usability of software interfaces, and the last two had no prior knowledge about modeling PVC supporting platforms. Each engineer filled in the survey and also a section that asks for their opinion regarding the pertinence of the considered NFRs. Using these results we tried to answer the stated research questions.

7.3 Obtained Results

The obtained results considered qualitative and also quantitative information. The next sections present these results and analyze the research questions based on that information.

7.3.1 Case Study 1: Bikers' Community

During the Formal Technical Reviews (FTR) performed to check the status of each project, the use of the proposed architecture helped the inspectors to identify relevant services that were not identified initially by the designers. During the stage 3 of the design process (i.e. architectural design) the product had several deficiencies. The most important were the following: there was no separation of the design concerns, and there was no support considered for *presence awareness* (I01, see Figure 2), *location awareness* (I02), *privacy settings* (U03), *feedback* (I07), *synchronous communication* support (I09), *content and media sharing* (I06) and structural services that support the community activities (i.e. those considered in the *community layer* – Figure 2).

The FTR performed at the stage 4 (i.e. detailed design) showed that various design limitations were addressed, and others were considered, but discarded by the designers because they wrongly decided that some services (like *feedback* (I07) and

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. They also manage how they will receive notifications (e.g. email, in-site).
Dashboard (U06)	A main page where the recent activity in the community, such as new messages and recent contributions are published automatically.
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.

Table 3. Requirements for a PVC Supporting System

content and media sharing (I06)) were not mandatory to support the community activities. The use of the architecture in this inspection helped designers realize the relevance of services considered in that model.

During the FTR conducted at the stage 5 (i.e. prototype implementation) the previous design limitations were reviewed. Provided that it was the last stage of the design process, the inspectors performed a general review of all services considered in the architecture. Four services were not included in the prototype; however all of them were opportunely evaluated by the designers. The *presence awareness* (I01) and *synchronous communication* support (I09) were not included because the designers decided to just use asynchronous communication as a way to keep persistent interaction among the community members. The other two services 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 designers because of time limitations. The design of this PVC supporting platform was graded with a 6.5 (in a range from 1 to 7), which indicates that it considers all services that are mandatory to support the activities of the community members.

Analyzing the results of the FTRs, which are documented in the corresponding inspection report, we can conclude that all services considered in the architecture were relevant in this project according to the designers' opinion. The decision to include or not, a certain service in the supporting system, was made considering other factors affecting the project, e.g. the available time to deliver the product or the level of criticism of such a service. This provides an initial positive answer to the RQ1.

In the interview performed to the inspectors after the last FTR, they manifested their intention of continuing the use of the architecture as a reference instrument during inspection activities for the next editions of the course. They indicated that the architecture allows them to easily keep in mind the most relevant design issues of a PVC supporting system and make visible the services that are not visible for the users. Thus, the architecture helps mitigate the iceberg effect affecting both, developers and inspectors. These preliminary results provide an initial positive answer to the RQ2.

The last inspection process (at the stage 5) also evaluated the usability and usefulness of the system according to the users' opinion. Therefore, the four potential users evaluated the services provided by the system. Both, the usefulness and usability of the system prototype, were rated as 'appropriate' to support the activities of the community. No extra services were identified by these potential users as mandatory for the supporting platform. This supports the previous answer for the RQ1.

7.3.2 Case Study 2: Players' Community

This project was developed in the same context as the previous experience; therefore it followed the same dynamics. The FTR conducted during the stage 3 (i.e. architectural design) showed results quite similar to the previous experience. For instance, the need for structural services (community layer) was not identified by the designers. Moreover, the design did not consider services to support *feedback* (I07), *user motivation* (U01), *presence* and *activity awareness* (I01 and I03), and *content and media sharing* (I06). The services identified by the designers were those that are visible through the user interface. This confirms the iceberg effect that usually affects developers of these system types.

The FTR performed at the stage 4 showed an important advance in the inclusion of the non-considered design aspects; however the design was still incomplete. The designers stated their intention to include services to address these design aspects, but they were not able to do it for this inspection because of time restrictions.

The FTR performed at the stage 5 showed that all design aspects specified in the architecture were embedded in the platform, but designers failed in 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 6.5 (in a scale from 1 to 7), and the students agreed on the limitations of their design to represent some design aspects. After performing the two reported projects, no additional services were identified as potential extensions to the proposed architecture. Preliminary, these results would indicate that the proposed architecture provides a good service coverage, and that the mandatory services are already considered in the architecture (RQ1).

The inspectors participating in this process were the same as in the previous case. They chose to continue using the architecture during the next editions of the course. This also reinforces the results obtained in the previous case related to the use of the architecture to evaluate product designs (RQ2).

Similar to the previous case, four potential users evaluated the usability and usefulness of the system prototype. They found that the services provided by the system were appropriate (RQ1); however, some of them, e.g. *presence* and *activity awareness* (I01 and I03) and also *content and media sharing* (I06), were not well implemented in the system.

7.3.3 Case study 3: Course Community

The evaluated prototype embedded implementations of all services considered in the proposed architecture. Such a product was used by 23 community members for eight weeks, and it was also reviewed by six engineers. Figure 7 shows the minimum, maximum and average values obtained through the survey given to both groups. These results indicate the usefulness of the proposed services according to users and designers. Dark bars indicate the average value assigned by the users to the *services usefulness*. Light bars represent *service usefulness* perceived by the designers (i.e. the engineers). A continuous scale from 0 to 5 was used to represent the usefulness of each service.

The results indicate that most services were considered as useful for the community members. Moreover, the usefulness assigned by the system users was quite similar to the ones assigned by the engineers. Analyzing the results and also the students' comments in the survey, we have identified some problems in the services implementation. Services like *synchronous communication* (I09) and *coordination* (I04) were not suitably implemented in the PVC supporting system. Therefore there is a significant gap between the expected and the perceived value of such services.

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

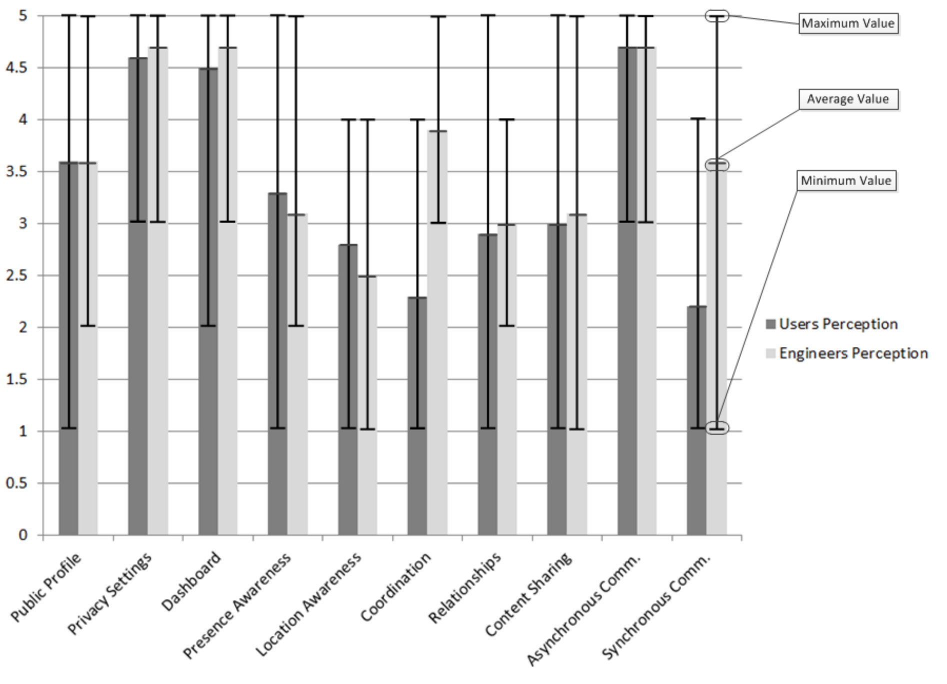


Figure 7. Services usefulness according to users and software engineers

Concerning the *coordination* service, the panel of software engineers recognized the importance of counting on this service, but the users assigned a usefulness value considerably lower than expected. This was also reflected in spontaneous comments that the users wrote at the end of the survey. These comments show a lack of initiative by 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 had been 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 not mandatory for any PVC supporting system. The relevance of these services depends on each particular case. This is aligned with the results obtained in the previous experiences. Therefore, it allows us to assume that the proposed services should be at least considered during the design process (RQ1). Provided that no services, additional to those already considered in the proposed architecture, were identified as mandatory, we can preliminarily assume that the proposal covers an important part of the design aspects involved in the PVC supporting systems.

The results also show us that some gaps between the expected and the real usefulness of a service can be present in these systems. The main causes 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).

The use of the architecture does not contribute to identify this type of system limitations. Analyzing the maximum, minimum and average values of the perceived usefulness of the services, we can realize that the dispersion of the users and designers opinions tends to increase if the supporting services are not well implemented.

The focus group conducted with software engineers resulted in similar results. According to the engineers' opinion, the FR and NFR were appropriately considered in the design of the system. These engineers also highlighted the simplicity of the software architecture, which makes it understandable and usable for many people. They were able to quickly understand the separation of concerns represented by the three-layer architecture, and also the role of the FR and NFR considered in the product design.

Five participants spontaneously pointed out that the designed platform lacked an effective support for activity awareness, as this feature would enhance participation between community members. This observation provides a preliminary positive answer to the RQ2, which is aligned with the previous results. Moreover, all designers considered this architecture so easy to learn and useful for analyzing PVC supporting systems in several contexts, and that reinforce the previous answer to the RQ2.

8 EVALUATING PVC SUPPORTING SYSTEMS

The designers participating in the previously reported case studies thought that the proposed architecture could also be used to evaluate already implemented PVC supporting systems. Therefore we defined a third research question:

(RQ3): How well does the proposed architecture support the evaluation of already implemented systems?

In order to obtain a preliminary answer to this question, three commercial PVC supporting systems were analyzed: *Facebook*, *U-Cursos* and *AcaMed*. In this analysis we attempted to verify whether or not those systems satisfy the set of requirements specified in the Section 4, and also if the non-addressed requirements are effectively required by the community members. Thus, we intended 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.

The authors have analyzed the first two PVC supporting systems (i.e. *Facebook* and *U-Cursos*) because they are active users of both platforms. Regarding the third one (i.e. *AcaMed*), two members of the development team independently conducted the analysis. In all cases, the evaluation process consisted of an informal inspection of the user interfaces and the identification of the structural components of each system. Then, we have matched these components with those considered in the proposed architecture. This helped us determine whether the services considered in the proposed architecture were considered in the product being inspected, and vice versa.

Moreover, we analyzed the relevance that each service seems to have in the inspected system. The service relevance can be rated as mandatory, recommended, desirable, dispensable, or negligible. Next we briefly present the informal analysis conducted for the three selected platforms.

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.

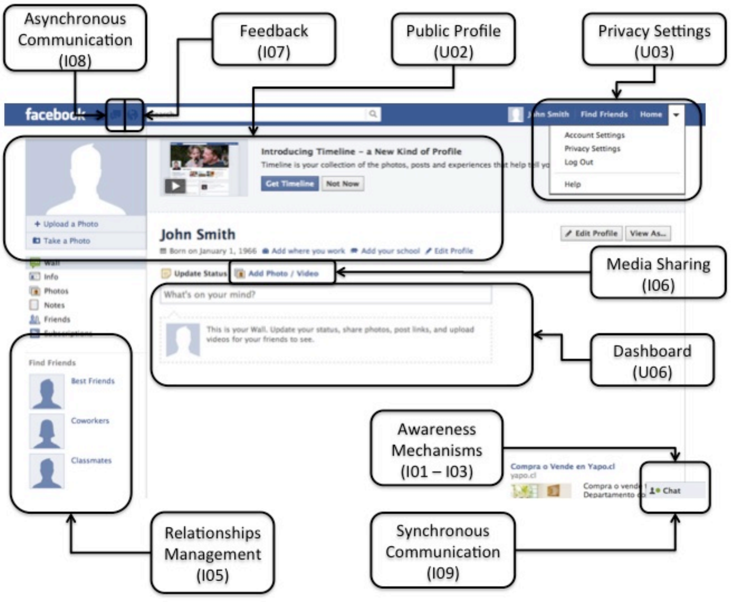


Figure 8. Facebook web user interface

We can see that most services considered in the architecture are part of Facebook. However, this platform lacks 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 the form of geo-tagging in status messages and in personal and shared pictures. This location awareness implementation limits the activities of a PVC, 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 (RQ3).

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*.

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 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. This provides a positive answer to the RQ3. 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 evalua-

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

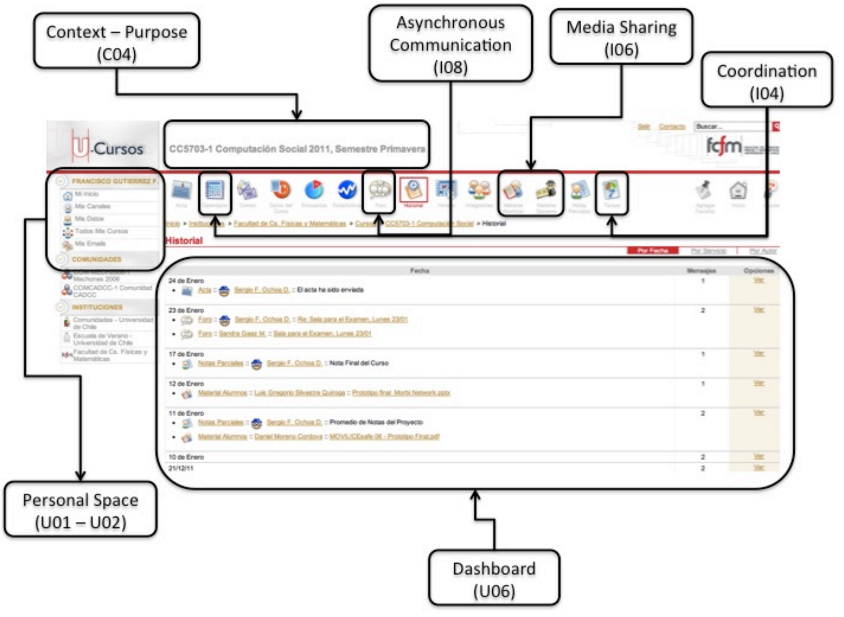


Figure 9. U-Cursos main user interface

tion period; therefore, they were not considered in this analysis. In other words, in this evaluation we considered only 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. The system also allows for managing members, membership fees and customizing the society 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, and 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.

According to the managers of this platform, its success relies mainly on the flexibility that it has to adapt the website not only to support the particular needs of a certain society, but also to evolve this support as the community evolves itself. When confronting the services available in *AcaMed* with those considered in

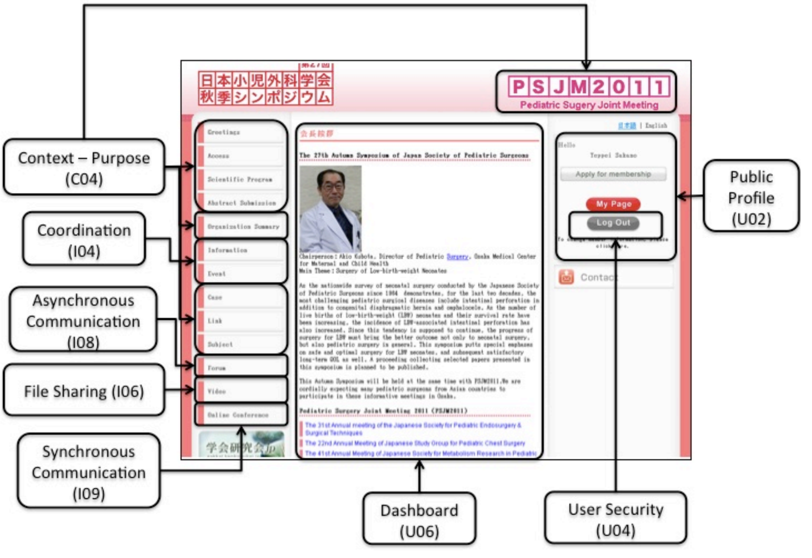


Figure 10. Main page of a website created using AcaMed

the proposed architecture we can see an important matching between them. 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*. We asked the system developers about the possibility to include these missing features. They answered that 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 was definitely a good idea, which was missing in the current implementation. About *security mechanisms*, they said that the data managed by the system was not sensitive 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.

Regarding the research question (RQ3), we can provide informal evidence supporting the use of the proposed architecture as an evaluation mechanism for already-implemented PVC supporting systems. In fact, software designers may be interested in filling out a traceability matrix where they can identify if the existing components of a particular software platform are considered or not in the current implementation of the PVC supporting system. Of course, the components described in our proposal are not always mandatory for any particular community. Therefore, software developers have to identify the particular needs of their communities, afterwards to verify their correspondence in the implemented platform, and finally conclude if the missing elements are actually required or are simply not relevant for the system.

In other words, the proposed architecture can be seen as a set of recommendations to be taken into account when designing, evaluating already-implemented, or envisaging possible modifications to PVC supporting systems. For example, in the latter case, as it was seen in *U-Cursos* and *AcaMed*, the components presented in the architecture can guide further explorations on how the platform may evolve in order to improve social interaction among community members.

9 CONCLUSIONS AND FUTURE WORK

According to Gutierrez et al. [23] a Partially Virtual Community is a hybrid between a physical and a virtual community. A PVC can be 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 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 is based on 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 interaction, and for the community), indicating their visibility level. 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 software platform. 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 interaction among members of a PVC

(*RQ1*). The architecture would also be useful to support the design of these systems (*RQ2*) and the evaluation of already implemented platforms (*RQ3*).

It is worth pointing out that our proposal is quite sensitive to the context in which the PVC supporting system is going to be designed. In fact, we do not aim to conceive an instrument that serves as a silver bullet for PVC supporting systems, but instead our goal is to provide software designers a set of guidelines to consider when designing and/or evaluating this particular kind of system. Also, we recognize that further exploration with software developers has to be taken into account, especially when analyzing if the proposal can actually be used as a tool for helping the evaluation of already-implemented PVC supporting system. It would also be interesting to discuss the pertinence of the proposal with regard to the evaluation of incremental functional prototypes, and even the applicability of the tool as a guide for design and evaluation of PVC supporting systems in agile software development teams.

The next steps in this research work consider evaluating the supporting platforms of the reported communities, but now relying on the help of their designers. This activity will provide us a better understanding of the strengths and weaknesses of this proposal because we will be able to completely address all the layers considered in the architecture. In addition, we 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 for their contributions to this work. This research work has been partially supported by the Fondecyt Project (Chile), Grant: 1150252; the LACCIR Project, Grant: R1210LAC002; and also the Internal Competition for Research Project Support – 2014, Facultad de Economía y Negocios de la Universidad de Chile. The work of Francisco J. Gutierrez has been supported by the Ph.D. Scholarship Program of Conicyt Chile (CONICYT-PCHA/Doctorado Nacional/2013-21130075).

REFERENCES

[1] Antunes, P.—Herskovic, V.—Ochoa, S. F.—Pino, J. A.: Structuring Dimensions for Collaborative Systems Evaluation. ACM Computing Surveys, Vol. 44, 2012, No. 2, pp. 8:1–8:28.

[2] Beenen, G.—Ling, K.—Wang, X.—Chang, K.—Frankowski, D.—Resnick, P.—Kraut, R. E.: Using Social Psychology to Motivate Contributions to Online Communities. Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04), Chicago, November 2004.

[3] Benghazi, K.—Noguera, M.—Rodríguez-Domínguez, C.—Pelegrina, A. B.—Garrido, J. L.: Real-Time Web Services Orchestration and Choreogra-

phy. Proceedings of the 6th International Workshop on Enterprise and Organizational Modeling and Simulation (EOMAS '10), Hammamet, June 2010.

[4] Bornoe, N.—Barkhuus, L.: Privacy Management in a Connected World: Students' Perception of Facebook Privacy Settings. Workshop on Collaborative Practices in Social Media, Hangzhou, March 2011.

[5] Buschmann, F.—Meunier, R.—Rohnert, H.—Sommerlad, P.—Stal, M.: Pattern-Oriented Software Architecture: A System of Patterns. John Wiley & Sons, Chichester, 1996.

[6] Chase, I. D.: Social Process and Hierarchy Formation in Small Groups: A Comparative Perspective. American Sociological Review, Vol. 45, 1980, No. 6, pp. 905–924.

[7] Cheng, R.—Vassileva, J.: Adaptive Reward Mechanism for Sustainable Online Learning Community. Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED '05), Amsterdam, 2005.

[8] Crandall, D.—Cosley, D.—Huttenlocher, D.—Kleinberg, J.—Suri, S.: Feedback Effects between Similarity and Social Influence in Online Communities. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '08), Las Vegas, 2008.

[9] Crumlish, C.—Malone, E.: Designing Social Interfaces. O'Reilly, Sebastopol, 2009.

[10] De Souza, C. S.—Preece, J.: A Framework for Analyzing and Understanding Online Communities. Interacting with Computers, Vol. 16, 2004, No. 3, pp. 579–610.

[11] Dellarocas, C.: The Digitalization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. Management Science, Vol. 49, 2003, No. 10, pp. 1407–1424.

[12] Dourish, P.—Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of the 1992 ACM Conference on Computer Supported Cooperative Work (CSCW '92), Toronto, 1992.

[13] Dunbar, R. I. M.: Neocortex Size as a Constraint on Group Size in Primates. Journal of Human Evolution, Vol. 22, 1992, No. 6, pp. 469-493.

[14] Dwyer, C.—Hiltz, S. R.—Widmeyer, G.: Understanding Development and Usage of Social Networking Services: The Social Software Performance Model. Proceedings of the 41st Hawaii International Conference on System Sciences (HICSS '08), Waikoloa, January 2008.

[15] Elgg Foundation Project: Open Source Social Networking Engine. Available on: `http://www.elgg.org`.

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

[17] Freedman, D. P.—Weinberg, G. M.: Handbook of Walkthroughs, Inspections, and Technical Reviews: Evaluating Programs, Projects, and Products. Dorset House Publishing, New York City, 2000.

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

[19] Garton, L.—Haythornthwaite, C.—Wellman, B.: Studying Online Social Networks. Journal of Computer-Mediated Communication, Vol. 3, 1997, No. 1.

[20] Girgensohn, A.—Lee, A.: Making Web Sites Be Places for Social Interaction. Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02), New Orleans, November 2002.

[21] Gonçalves, B.—Perra, N.—Vespignani, A.: Modeling Users' Activity on Twitter Networks: Validation of Dunbar's Number. PLoS One, Vol. 6, 2011, No. 8, DOI 10.1371/journal.pone.0022656.

[22] Gutierrez, F.—Baloian, N.—Zurita, G.: Boosting Participation in Virtual Communities. In: Vivacqua, A. S., Gutwin, C., Borges, M. R. S. (Eds.): Collaboration and Technology, Proceedings of the 17th International Conference (CRIWG '11), Paraty, October 2011.

[23] Gutierrez, F.—Baloian, N.—Ochoa, S. F.—Zurita, G.: Designing the Software Support for Partially Virtual Communities. In: Herskovic, V., Hoppe, H. U., Jansen, M., Ziegler, J. (Eds.): Collaboration and Technology, Proceedings of the 18th International Conference (CRIWG '12), Raesfeld, September 2012.

[24] Gutwin, C.—Greenberg, S.: Effects of Awareness Support on Groupware Usability. Proceedings of the 1998 ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '98), Los Angeles, April 1998.

[25] 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, 2011, No. 2, pp. 183–202.

[26] Herskovic, V.—Neyem, A.—Ochoa, S. F.—Pino, J. A.—Antunes, P.: Understanding Presence Awareness Information Needs Among Engineering Students. Proceedings of the 16th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD '12), Wuhan, May 2012.

[27] Hill, T.—Supakkul, S.—Chung, L.: Confirming and Reconfirming Architectural Decisions on Scalability: A Goal-Driven Simulation Approach. Proceedings of the OnTheMove Federated Conferences & Workshops (OTM '09), Vilamoura, 2009.

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

[29] 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, 1989.

[30] Hunter, M. G.—Stockdale, R.: Taxonomy of Online Communities: Ownership and Value Propositions. Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS '09), Waikoloa, January 2009.

[31] Iriberri, A.—Leroy, G.: A Life-Cycle Perspective on Online Community Success. ACM Computing Surveys, Vol. 41, 2009, No. 2.

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

[33] 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, 1996, pp. 254–384.

[34] Kollock, P.: Design Principles for Online Communities. Proceedings of the Harvard Conference on the Internet and Society, 1996.

[35] Lampe, C. A. C.—Ellison, N.—Steinfield, C.: A Familiar Face(book): Profile Elements as Signal in an Online Social Network. Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '07), San Jose, May 2007.

[36] Lee, A.—Danis, C.—Miller, T.—Jung, Y.: Fostering Social Interaction in Online Spaces. Proceedings of the IFIP International Conference on Human-Computer Interaction (INTERACT '01), Tokyo, 2001.

[37] Lee, F. S.—Vogel, D.—Moez, L.: Virtual Community Informatics: A Review and Research Agenda. Journal of Information Technology Theory and Application, Vol. 5, 2003, No. 1, pp. 47–61.

[38] Liu, Y.—Gummadi, K. P.—Krishnamurthy, B.—Mislove, A.: Analyzing Facebook Privacy Settings: User Expectations vs. Reality. Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement (IMC '11), Berlin, November 2011.

[39] Lobry de Bruyn, L.: Monitoring Online Communication: Can the Development of Convergence and Social Presence Indicate an Interactive Learning Environment? Distance Education, Vol. 25, 2004, No. 1, pp. 67–81.

[40] 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, 2006, No. 6, pp. 1014–1031.

[41] McMillan, D. W.—Chavis, D. M.: Sense of Community: A Definition and Theory. Journal of Community Psychology, Vol. 14, 1986, No. 1, pp. 6–23.

[42] McPherson, L.—Smith-Lovin, L.—Cook, J.: Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology, Vol. 27, 2001, pp. 415–444.

[43] Menascé, D. A.—Almeida, V. A. F.: Capacity Planning for Web Services: Metrics, Models and Methods. Prentice Hall, Upper Saddle River, 2001, pp. 415-444.

[44] Mislove, A.—Viswanath, B.—Gummadi, K. P.—Druschel, P.: You Are Who You Know: Inferring User Profiles in Online Social Networks. Proceedings of the 3rd International Conference on Web Search and Data Mining (WSDM '10), New York City, February 2010.

[45] Mousavidin, E.—Goel, L.: A Life Cycle Model of Virtual Communities. Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS '09), Waikoloa, January 2009.

[46] Neyem, A.—Ochoa, S. F.—Pino, J. A.: A Patterns System to Coordinate Mobile Groupware Applications. Group Decision and Negotiation, Vol. 20, 2011, No. 5, pp. 563–592.

[47] Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco, 1993.

[48] Parameswaran, M.: Social Computing: An Overview. Communications of the Association for Information Systems, Vol. 19, 2007, pp. 762–780.

[49] Plant, R.: Online Communities. Technology in Society, Vol. 26, 2004, pp. 51–65.

[50] Porter, C. E.: A Typology of Virtual Communities: A Multi-Disciplinary Foundation for Future Research. Journal of Computer-Mediated Communication, Vol. 10, 2004, No. 1, DOI 10.1111/j.1083-6101.2004.tb00228.x.

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

[52] Preece, J.: Sociability and Usability: Twenty Years of Chatting Online. Behavior and Information Technology Journal, Vol. 20, 2001, No. 5, pp. 347–356.

[53] Preece, J.—Maloney-Krichmar, D.: Online Communities. In: Jacko, J., Sears, A. (Eds.): Handbook of Human-Computer Interaction. Lawrence Erlbaum Associates Inc., Mahwah, 2003.

[54] Preece, J.—Shneiderman, B.: The Reader-to-Leader Framework: Motivating Technology-Mediated Social Participation. AIS Transactions on Human-Computer Interaction, Vol. 1, 2009, No. 1, pp. 13–32.

[55] Ramsey, D.—Beesley, K. B.: 'Perimeteritis' and Rural Health in Manitoba, Canada: Perspectives from Rural Healthcare Managers. Rural and Remote Health Vol. 7, 2007, pp. 850.

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

[57] Sonnenbichler, A. C.: A Community Membership Life Cycle Model. Proceedings of the XXIX INSNA International Sunbelt Social Network Conference, San Diego, 2009.

[58] Supakkul, S.—Hill, T.—Chung, L.—Tun, T. T.—Leite, J. C. S. P.: An NFR Pattern Approach to Dealing with NFRs. Proceedings of the 18^{th} IEEE International Requirements Engineering Conference (RE '10), Sydney, September 2010.

[59] Tedjamulia, S. J.—Olsen, D. R.—Dean, D. L.—Albrecht, C. C.: Motivating Content Contributions to Online Communities: Toward a More Comprehensive Theory. Proceedings of the 38^{th} Hawaii International Conference on System Sciences (HICSS '05), Big Island, January 2005.

[60] 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.

[61] 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, 1999, No. 4, pp. 587–599.

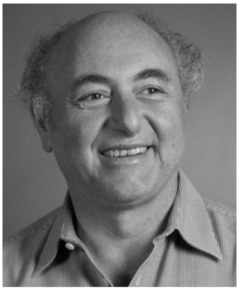
[62] Walther, J. B.: Computer-Mediated Communication: Impersonal, Interpersonal, and Hyperpersonal Interaction. Communication Research, Vol. 23, 1996, No. 1, pp. 3–43.

[63] Westerlund, M.—Rajala, R.—Nykänen, K.—Järvensivu, T.: Trust and Commitment in Social Networking – Lessons Learned from Two Empirical Studies. Proceedings of the 25^{th} IMP Conference, Marseille, 2009.

[64] Xu, L.—Ziv, H.—Richardson, D.—Liu, Z.: Towards Modeling Non-Functional Requirements in Software Architecture. In: Workshop on Aspect-Oriented Requirements Engineering and Architecture Design, Chicago, March 2005.

Francisco J. Gutierrez is a Ph.D. candidate in computer science at the University of Chile. His research interests include human and social factors in computing systems, social computing, computer-mediated communication, mobile and ubiquitous computing, and computer science education. He is a student member of ACM, and of the ACM Special Interest Group on Computer-Human Interaction (SIGCHI).

Sergio F. Ochoa is Associate Professor of computer science at the University of Chile. He received his Ph.D. in computer science from the Catholic University of Chile. His research interests include computer-supported collaborative work, software engineering, mobile and ubiquitous computing, and educational technology. He is a member of IEEE, ACM and the Chilean Computer Society, and a member of the Steering Committee of the LACCIR (Latin American and Caribbean Collaborative ITC Research Initiative). He has worked as a software engineer for more than ten years and he currently serves as an IT consultant for a number of public and private organizations in Chile.



Nelson Baloian graduated in computer engineering from the Universidad de Chile in 1987. He received his Ph.D. from the University of Duisburg, Germany in 1997. His main research subjects were computer supported learning and distributed systems, he is author of more than 20 journal publications related to these subjects and has peer reviewed 80 conferences. He was Visiting Professor at the universities of Waseda, Japan, and Duisburg-Essen, Germany. He is currently Associate Professor at the Department of Computer Science of the Universidad de Chile in Santiago, Chile.

Gustavo Zurita received his Ph.D. in civil engineering, with computer science major, from the Catholic University of Chile. Currently he is Associate Professor at the Department of Management Control and Information Systems of the Faculty of Economics and Business of the Universidad de Chile. His main research topics include computer supported mobile collaborative learning, collaborative knowledge construction and geo-collaboration. He is author of about 50 conference articles and more than 20 ISI-indexed journal articles. Some of his journal articles are among the most cited ones in mobile computer supported learning in the world.

Luis Loyola obtained his M.Sc. and Ph.D. on Information Systems and Telecommunications at the University of Electro-communications in Tokyo, Japan. He worked two years as a research engineer at the Research and Development division of NTT in Japan, and then three years at NTT Docomo Europe Labs in Munich, Germany. During his time at the NTT group, he worked on the study and design of new resource allocation and transmission schemes for advanced wireless systems, cooperative coding in wireless channels, wireless networks multicast, peer to peer networks, network coding for live streaming applications and file sharing, multihop and ad-hoc network performance analysis, QoS over multi-hop networks and multi-hop multi-radio networks. In 2007 he received a prize at NTT Docomo Europe Labs for his outstanding contributions to the company. From August 2008 he is leading the R & D division at SkillupJapan Corporation in Tokyo, Japan, working on Partial-Reliability SCTP extensions for H.264 video streaming, task parallelization, network coding for forward error correction in media streaming, distributed storage, recommender engine algorithms, noSQL databases, big data analytics and data visualization techniques.