Simulated Architecture and Programming Model for Social Proxy in Second Life

Cintia Caetano, Micheli Knechtel, Roger Resmini, Ana Cristina Garcia, Anselmo Montenegro
Department of Computing, Fluminense Federal University, Niterói, Rio de Janeiro, Brazil
ccaetano@ic.uff.br, micheliknechtel@gmail.com, rresmini@ic.uff.br, bicharra@ic.uff.br, anselmo@ic.uff.br

Abstract

This paper presents an approach for developing a Social Proxy applied to the Collaborative Virtual Environment (CVE). Also are described the architecture, programming model and evaluation strategies. Our Social Proxy is called SP HUD, a component that provides a view in third person allowing a clearer view of the group activities. With it, we can visualize all the avatars presents in the shared space and to know the interaction forms of each one in SP HUD. We can describe three interaction forms. The first form describe that the user does part the group, the second form describe the avatar´s presence in space and the third form describes that the avatar is interacting with other one. To build the SP HUD component we used a known CVE, called Second Life® which is based on client-server architecture, and uses a centralized-primaries model architecture. The social proxy is a component that is part of a larger project and to test the level of acceptance of the user on the quality of the application developed, we perform six experimental meetings.

1. Introduction

A Collaborative Virtual Environment (CVE) is a space for interaction that supports construction, integration and exchange of information from many participants who may eventually be spread over great distances. This space can be used in many different forms of interaction such as performing of tasks, meeting targets, construction of knowledge, social networking, etc.

The abundant amount of data and the granularity of the interactions of CVEs has stimulated numerous investigations on forms of user cooperation and collaboration.

The richness and variety of multimedia applications, characterized by processing of large data volumes and continuous play time, have attracted the interest of the CSCW community (Computer Supported Cooperative Work) for some time. CSCW is defined as the area of research that studies the use of computer and communication technologies to support activities of groups and organizations [1].

To support this large power of multimedia interaction and communication, CVEs are developed based on an architecture (computational model), which greatly influences the use of them. This work demanded a research about the available architectures, as well as their external properties, and their relationships with others softwares, etc.

In this paper we describe the process of developing a CVEs application, called Social Proxy, including the used architecture, the programming model and the evaluation strategies. The Social Proxy is a minimalist visualization of the presence and activities of participants in online interactions [2]. It is used to support collaboration in meetings of knowledge acquisition and decision making.

The CVE selection was based on preliminary studies on the limitations and available resources for the development of Social Proxies. As a result of such investigation the Second Life was selected as the CVE used for the development of the application. Second Life is a 3D virtual environment where players develop strong social bonds interacting with objects, forming groups and participating in activities of significant value [3][4][5].

Second Life bears for different study possibilities, providing subsidies for users to make inferences about the activities of the members of a group and capture the nuances of their behavior.

2. Social Proxy

Many problems may be triggered when a work is done collaboratively through shared computer applications as interaction between users, concurrency issues and others etc. In an essay to solve these problems, signals are used to alert the user, such as informational messages which become invisible to the users as time evolves. Other attempts as making users visible to each other are also implemented, but when visibility is present, the social rules have to be imposed.

When we are at the mercy of social rules, we must be prepared for different types of behavior. User A for example may have an unusual behavior or even be aggressive.

The visible status of a user allows others to perceive their degree of interaction within the group at a given time and also allows a look at the User level of mutual interaction making sure that everyone is aware of the context and the existing activities of the other members, also acknowledging those who are responsible for the actions.

Based on the problems of visualization and use of collaborative applications, Thomas Erickson [2] designed the Social Proxy, a minimalist view of the presence and activities of participants in online interactions. He has an approach where social norms coordinate the collective behavior, causing physical and behavioral signs visible within the online environment.

The virtual scenario is represented graphically by a circle of diameter larger than a set of dots, which represent the users in the environment. Each point is assigned a unique color in such a way that it is possible to get information about a particular individual by identifying its color.

The disposition of points (far or near the center circle) represents the configuration of user interaction with each other. The closer to the center of the circle the greater is the degree of interaction between them.

The figure 1 represents the Social Proxy interface that was implemented for a system multi-room persistent chat environment called Babble [6].

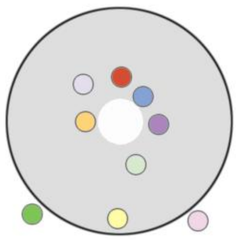

Figure 1. Social representation of the a chat group in the Babble system

When a participant reads, or speaks, his point is moved to the center circle. When no longer active, his point moves to the periphery of the circle. [2]

It is natural that during face-to-face activities people quickly starts to perceive the environment and its changes. In virtual environments, the support for perception is less clear, because the means of information transmission to the sensory organs of humans are restricted [7].

The point of greatest relevance of the Social Proxy is certainly the awareness, that is, the form in which knowledge about the activities of a particular individual or group is available to the user [8].

3. Second Life Environment

Second Life is a virtual environment in real time, shared among its users. Within this universe, which simulates real life, users are controlled by remote players through their virtual representation, commonly called avatars. Everything that happens can be seen by all users immersed in it, so that the information is conveyed at once to many people.

This environment has several characteristic, enabling members of the groups to be together, for example, in virtual meeting rooms, simulating a meeting in a physical space of the real world, which allows greater immersion in collaborative activities.

All interactions has full support for voice (synchronous) and text (synchronous and asynchronous), as well as the use of gestures to enhance communication and therefore the immersion.

Besides interacting with other players, Second Life users interact with the environment itself, which has a variety of objects, animations and properties that can give him the ability to optimize the avatar and his way of “living” in the world.

Second Life allows users to recreate and expand the current social bonds. According to Erickson [2], 3D virtual environments are a great approach to represent the social norms available, as they allow the reconstruction of the physical signs present in the real world, and the participants can produce and respond to suggestions for behavior by manipulating their avatars.

Groups working together can produce better and faster than those which work individually. Perceive the nuances of behavior of a virtual group is not an easy task in a virtual environment, even though crucial to the performance of tasks among members of a group.

Based on the six guidelines for developing the Social Proxy listed by Erickson [2] we developed a version of Social Proxy for Second Life, in order to support collaborative meetings allowing each member to have a view of the behavior of their group, both in a global or local scale.

The central idea is to provide the participant of collaborative meetings a clearer visualization of the activities of a member of the group and the activities of the group as a whole.

Through basic primitives of the environment and using resources provided by the LSL language (native to Second Life) we developed a 3D component, called SP HUD that is able to represent the nuances of behavior of a group.

The SP HUD is updated dynamically and is visible to all the avatars that are within a predetermined area.

The application project takes into account the awareness information, as they can be obtained or generated, where elements of awareness are needed, how to present them and how to give individuals control over them.

There was also a large concern about excessive information, not to cause overload and hinder their collaboration.

4. Architecture and Programming Model

The CVEs applications are usually based on the shared virtual environment. Because of the spreading of participants and the communication latency, some data consistency model has to be used to keep the data consistent [9].

The CVEs architecture influences deeply the programming model of the application. Usually the distributed systems architecture classification is based on several criteria, like centralized/distributed architecture, type of replication, performance and consistency properties [10].

The most frequently used CVE architectures [10] can be classified in four types: centralized primaries, all primary replicas of each data item resides in the server; distributed primaries, primary replicas are distributed among the computers; data ownership, primaries are allowed to migrate among the computers and active replication, based on peer-to-peer approach in which all replicas are equal. Usually, atomic broadcast is used to deliver updates to all of them, thus they are kept synchronized.

The Collaborative Virtual Environment used to developing the Social Proxy application is Second Life which is based on client – server architecture. in order to solve problems arising from the architecture and infrastructure of the environment, especially the difficulty of maintaining the consistency of the data due to latency and bandwidth, we simulate the centralized primaries model architecture in Social Proxy Application.

The social proxy application is composed by two components, first the SP's head-up display (HUD), that is attach by the user in the interface for visualization of the group behavior, and second one invisible object for the user, called radar.

Each SP HUD represents a network node, and maintains a replica of the scenery data. The invisible radar object represents a network node, located in the environment, where the consistent information of the scenery is kept, updated and distributed. In other words, it holds the primary copy of the data item.

The figure 2 is an example of schematic simulation of the centralized primaries model architecture in Social Proxy Application.

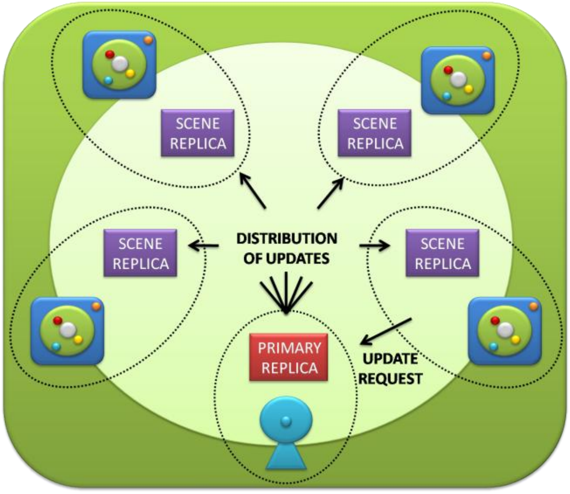

Figure 2. Model architecture in Social Proxy application

Every change in the user behavior is detected by his own SP HUD, and then updated by exchanging messages between the SP HUD and radar.

The radar responds to changes of scenery notification, updates the primary copy and distributes the replica of the current scenario to the SP's HUD by data broadcasting.

There are three behaviors that can trigger changes in the scenery: user on-line, off-line and typing. The figure 3 is an example of three behaviors that Social Proxy can represent.

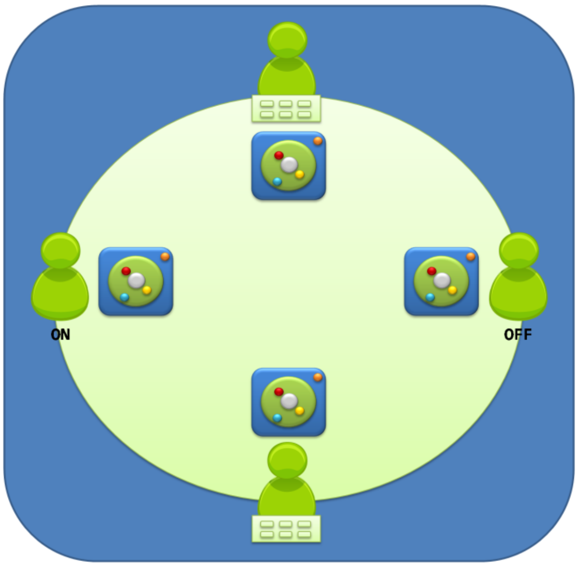

Figure 3. Snapshoot of current scenery

5. Analysis

In order to obtain the level of acceptance of the user on the quality of the application developed, we held six experimental meetings, within about two hours each, whence the participants (volunteers) had the opportunity to interact and observe the behavior of the group to which they belong, through the Social Proxy.

The figures 4 and 5 show a screenshot of one of the participants of the experimental meetings.

Figure 4. Overview of Social Proxy at the meeting

Figure 5. Avatars interacting during the meeting

Participants reported that during the meeting the Social Proxy proved to be an informative and useful component, because it facilitates the awareness of who is at the meeting and if they are participating.

The Social Proxy presented some limitations, for example, cannot be scaled to large numbers of people, because the minimalist visualization does not allow the view of large groups. Despite its visualization being very easy to interpret, the SP HUD occupies a significant quantity of screen space.

The Social Proxy is able to display only if someone is typing, if someone is online or offline. As supposed by Thomas Erickson things that users "see/interpret" in the social proxy are not strictly correct, because users are making inferences from the visualization. If an avatar is not typing, this is not to say that he is not in the conversation, he may be just paying attention. These mistaken inferences may go unnoticed or can be easily detected.

Considering the learning problems, the experiments have shown that all participants could understand the fact that "some event was happening" when a cluster of balls near the center circle was displayed.

Participants were solicitous in teaching the others members about the interpretation of signs. Knowledge sharing occurred naturally and very often. Participants usually helped each other when some problem occurred.

The development within Second Life is not a simple task, because the scripting language has many limitations and restrictions that hinder the development. In these cases, we had to find ways to overcome limitations and achieve the desired results.

Communication between components is a clear example, because communication between objects is done by sending messages. To implement the synchronism manually is a very cumbersome task, which could be simplified considerably if the language were more expressive and robust.

The edition of shared scripts is not supported by Second Life, thus the implementation of the scripts was done in single-user mode. The language is still very limited; it has no arrays, switch case, matrices, and other structures common to conventional programming languages.

Infrastructure issues, especially the bandwidth, are also factors that hinder development. As it does not provide an "off-line" mode, all the development in Second Life must be done through client/server communications, that is to say compiling and updating the code on the server whenever a new change must be done. Also the need to use the Internet as means of interaction with objects, creates problems that hides errors and make components difficult to test.

6. Conclusions

Perceive the activities of other individuals is essential to ensure the flow and naturalness of the work so as to lessen the feelings of impersonality and distance common in virtual environments. In this proposal we developed a Social Proxy for Second Life which acted of form simple and easy, providing perception without burdening the users' activities.

As part of a larger project, our Social Proxy Application has been tested at the level of acceptance, which allowed us to verify the implementation and architecture used. For future work, we plan to conduct controlled experiments with user groups in collaborative activities in order to achieve more tangible results.

Although the limitations of Second Life and extra effort to avoid them, the end result was satisfactory. With the improvement of CVE's and their applications, it is expected that the interaction becomes increasingly simple and pleasant, attracting a growing number of users.

7. References

[1] Olson, G. M. and Olson, J. S. (1997). Research on computer supported cooperative work. Handbook of human-computer interaction, 2nd. Ed., Helander, M.G., Landawer, T.K., Prabhu, P.V. (ed.).

[2] Erickson, T. 2008. ‘Social’ systems: designing digital systems that support social intelligence. AI Soc. 23, 2 (Aug. 2008), 147-166.

[3] Brown, B. and Bell, M. 2004. CSCW at play: 'there' as a collaborative virtual environment. In Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work CSCW '04. Chicago, Illinois, USA, November 06 - 10, 2004.

[4] Ducheneaut, N., Wen, M., Yee, N., and Wadley, G. 2009. Body and mind: a study of avatar personalization in three virtual worlds. In Proceedings of the 27th international Conference on Human Factors in Computing Systems CHI '09. Boston, MA, USA, April 04 - 09, 2009.

[5] Olivier, H., & Pinkwart, N. (2007). Computer Supported Cooperative Work In The Second Life? Position paper presented at the Track on Business in Second Life at the 3rd International Second Life Community Convention. Chicago.

[6] Erickson T, Smith DN, Kellogg WA, Laff MR, Richards JT and Bradner E Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of Babble. In The Proceedings of CHI 1999. ACM Press, New York.

[7] Gerosa, M., Fuks, H. e Lucena, C. Suporte à Percepção em Ambientes de Aprendizagem Colaborativa. Revista Brasileira de Informática na Educação – V.11 N. 2 – 2003

[8] Pinheiro, M.K.; Lima, J.V.; Borges, M.R.S. (2001). “Awareness em Sistemas de Groupware”. In: International Database Engineering and Applications Symposium, 2001, San Jose, Costa Rica. Proceedings... abr. 2001. p. 323-335.

[9] Bouras C., and Filopoulos A. 1998. Distributed Virtual Reality Environments Over Web for Distance Education. In: http://ru6.cti.gr/Publications/269.pdf. Access date 15 December 2009.

[10] Lavery, E. (2008) Introduction to Flexsim DS. Flexsim Corp. In: http://www.flexsim.com. Access date 12 December 2009.