CSCW Systems on PvC Environments

Andrés Flores
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
Buenos Aires 1400, Neuquén, Argentina
phone: (54) 299 - 4490312, fax: (54) 299 4490313
Email: aflores@uncoma.edu.ar

Abstract. CSCW is a well-suited discipline for appropriate communicating ideas inside a group and even in an inter-group cooperation. Though basic technology for communication annoy people with too many computer-related tasks to deliver at-the-time elaborated notes in the meanwhile of a group activity. An emerging trend that promises to bring a solution on this matter is that of ubiquitous, or pervasive computing. PvC is about computation becoming part of the environment with the ambition to accomplish accessing information anytime no matter the distance of user's location. This is not only related to largely distributed systems and applications, but about highly dynamic and mobile sets – clusters of participants, interacting with each other and storing data on mobile devices, as well as in remote facilities. PvC systems certainly come to facilitate CSCW making every kind of such systems to look as the simplest groupware option and providing a new and particular kind of such systems, what might be called as PvCE-SCW. This paper reports on current efforts on this type of systems describing their requirements and challenges with the intent to provide a summary of successful accomplishments on this matter.

Keywords: CSCW, Pervasive Computing Environments.

1. Introduction

CSCW systems have become a place for proper communication on collaborative environments. According to the groupware typology different technologies have been used to give support to implement its corresponding system. Some results adequate for more than one very particular groupware activity, others are too narrow in its utilization. Most of them, however, bring additional negative effects on others but the target activity for its deployment. The interdependence among activities has given a warning for developers not to focus attention too narrowly but going for an ultimate intent: "any time, any place" support [1]. Ubiquitous, or pervasive computing is an emerging trend that promises to give a solution on this matter. Pervasive computing (PvC) is about computation becoming part of the environment, a computing universe populated by a rich variety of heterogeneous computing devices: smart cars, entertainment systems, appliances systems, where many different protocols and operating systems are interrelated with the ambition to accomplish data to be delivered efficiently, effectively, and economically to any device, no matter the time or distance of user's location. This is not only related to largely distributed systems and applications, but about highly dynamic and mobile sets – clusters of participants, interacting with each other and storing data individually on mobile devices, as well as in remote facilities [2,3].

The major goal is to achieve an implicit human-computer interaction. That is to render computing devices and their technology invisible so the user remains focused on the matter at hand: receiving updates and tasks. The traditional nature of user's relationship to computation changes in a pervasive environment. The difference is the explicitness of the computational task, where people think in terms of performing explicit tasks "on the computer" – creating documents, sending e-mail, and so on. This model of interaction responds to a low level abstraction: individual applications and devices. When computation is part of the environment this explicitness disappears as individuals behave as they normally do: moving around, using objects, seeing and talking to each other. The computation in the environment is in-charge of facilitating these actions, and individuals may come to expect certain services to achieve high-level goals – a high-level task interaction [2,4,5].

PvC systems certainly have come to provide a helpful support for CSCW systems. They make every kind of system on taxonomy [1,6] to be seen as the simplest groupware option. In fact we may consider collaborative systems under pervasive computing environments as a special kind of system: pervasive-computing-environment-supported-collaborative-work systems or PvCE-SCW systems. The reason to this name is the goal under PvC environments, not to have a computer-oriented task consciousness but a high level task interaction as was pointed out before.

Developing software for such systems is a quite challenging activity. There are numerous and conflicting requirements to be considered, so an adequate analysis of current efforts on this matter could be the necessary step to follow. Thus we may achieve an overview of concrete needs and their possible solutions. From the knowledge of proven solutions developers may improve their possibilities to accomplish a better construction of such necessary but challenging systems. In section 2 a groupware typology and CSCW system requirements are introduced. They will be seen at the light of PvC systems presenting a report on current efforts for PvCE-SCW systems recognizing solutions to those requirements depicted in the previous section. Additionally, commonalities among solutions will be highlighted, which tell about recurring proven solutions. Then section 3 presents conclusions and future work.

2. PvCE-SCW Systems

There is a recurring question about what should be included under the rubric of *groupware* or *CSCW applications*. In [1] is given a familiar groupware typology based on a multidisciplinary perspective. Figure 1 shows a categorization based on two edges: place and time. Thus one activity can be carried out in a single place (top row); in several places known to the participants, as in electronic mail exchanges (middle row); or in numerous places, not all of which are known to participants, as in a message posted to a netnews group (bottom row). Activity can be carried out *in real time* – that is, in one unbroken interval, as in a meeting (left column). Alternatively, it can be carried out at different times that are highly predictable or constrained, as in sending mail to a colleague and expecting it to be read within a day or so (middle column). Or it can be carried out at different times that are unpredictable, as in open-ended collaborative writing projects (right column).

		Time		
		Same	Different but predictable	Different and unpredictable
Place	Same	Meeting facilitation	Work shifts	Team rooms
	Different but predictable	Tele/video/ desktop conferencing	Electronic Mail	Collaborative writing
	Different and unpredictable	Interactive multicast seminars	Computer bulletin boards	Workflow

Figure 1: Map of groupware options, categorizing representative applications

Activities do not always match Figure 1 precisely. Most real work does not fall into one or another category. As we go about our work, we generally engage in some face-to-face meetings and some distributed and asynchronous communication. Most involves both communication and coordination. Narrow tasks interact with broader work activities, and even the broadest concerns overlap and impact one another.

Technology designed to support activity in one cell can fail by negatively impacting activity in another. For example, a standalone meeting support system that provides no access to existing databases or other on-line materials may be useless in some situations. It is notable the interdependencies among activities, which may tell us to think about a real need for *any time, any place support*. This last requirement is certainly the goal of PvC systems, and so they can be helpful by facilitating communication under group-aware features inherent to such systems.

In order to better understand the needs on CSCW systems and clarify the reasons why PvC systems are a promising solution we following depict concrete requirements under specific criteria according to [6]. They can be recognized into the previous map given above and will highlight those aspects that can be improved through the usage of PvC systems. Thus PvCE-SCW systems will be properly justified as a new particular kind of collaborative systems.

Functional criteria: A CSCW system relates functional features with the social aspects of teamwork. Each functionality has an impact on the work behaviour and efficiency of the entire group using the system. These functionalities also influence the behaviour of individual group members. However, the psychological, social, and cultural processes active within groups of collaborators are the keys to the acceptance and success of CSCW systems.

Interaction. It can proceed *synchronously* or *asynchronously*, depending on the required response time. Joint editing of the same text is done synchronously, whereas sending e-mail is an asynchronous task. Interaction can also be classified as *implicit* or *explicit*. The former relates to the object of joint work (such as shared text or images). The latter deals with direct communication among coworkers through gestures, voice, and video transfer. Interaction can be *formal*, if it relies on formal procedures, or *informal*, as in a joint session of equal partners. In [9] DISCIPLE presents a synchronous interaction and is classified as informal and explicit since it provides tools for communicating ideas instead for working on a shared piece of work. In [10] MERBoard intents to present a formal interaction by formal meeting process; where the system reaction over users creation of something is a document that becomes formalized. In [11] WebSplitter is rather synchronous in its interaction by a explicit and informal way of communication.

Coordination. Coordination of interaction deals with communication within the group of users; it depends on the group's size and the way individual group members prefer to interact. Small groups usually need less coordination than large groups. Therefore, interaction modes change with the group's size and task. A *free mode* relies on agreement among the participants, while a *system-based mode* can be provided by the CSCW system. In [9] DISCIPLE allows new members be integrated to the group dynamically with the proper setting on their roles. Additionally, there are not constraints at fixing and changing member roles dynamically. MERBoard in [10] is being developed under a system-based mode as well. WebSplitter in [11] also corresponds to a system-based mode of interaction. In [12] a big effort has been done on information consistency among group members on the Apia approach.

Distribution. Computers enable people to interact from remote places. A distributed working environment requires additional transfer channels for explicit interaction involving speech and gestures. Besides incurring technical problems, global distribution of cooperating partners crosses time zones and borders that pose social, cultural, and political differences (for example, languages, negotiation strategies, behaviours, metrics, and laws). DISCIPLE in [9] relies on the Internet which lets its users to be widely disparate. In order not to affect users with complicated ways of communication as those presented in the map of groupware above, this approach is base on a simple session as a meeting place. The approach in [11] allows distributed users under Internet infrastructure as well.

User-specific reactions. One approach for building a CSCW system is to distribute an existing single-user application via an application-sharing system. In such a system, each interaction is handled the same way regardless of individual users' roles. CSCW systems without user-specific reaction are *collaboration transparent.* For joint work, this type of system lets users work with a

familiar user interface and access existing, unmodified application programs. Certain users sometimes desire specific system reactions, depending on their individual roles or rights. This kind of CSCW system is *collaboration aware*; it "knows" the number of users and their individual roles, and is specifically developed as a multiuser application for cooperative work. DISCIPLE in [9] includes a *collaboration space* which is a directory of all people that a user can collaborate with. This workspace can be represented in several ways and is provided a model of management. Thus it is certainly for collaboration aware, though is too limited for providing collaboration transparent. In [11] the approach is also classified as collaboration aware since it keep a record of every user and its roles, preferences and privileges.

Visualization. The what-you-see-is-what-I-see paradigm determines how to visualize public data used collaboratively. Pure WYSIWIS is strongly related to collaboration transparent CSCW systems. There are several levels of relaxed WYSIWSIS. Users might see screens with individual layouts but identical content in shared data (such as different chapters of the same document in a shared editor). The application itself can provide individual views (different colours, cursor shapes, or different shared-data-layouts). In [9] it is possible for different users to personalize their views of the group and places. This approach is feasible since it is not collaboration transparent. In [11] public data is delivered according to available appliances for users by splitting information among more than one device if necessary.

Data hiding. Nobody would use CSCW if it did not contain a way of separating private from public data. Public data should be classified as such and should be accessible to certain persons, groups, or the public in general. The granularity of data should not be restricted to file level, but to finer information units like data objects. In [9] many different elements of data may be managed and interchanged. In addition, changing the mode of workspaces inside the group gives privacy for users. In [11] there is an effort on information hiding according to group members' roles and privileges.

Technical criteria: comprises hardware, software, and network support. *Hardware* criteria refer to the target machine and I/O devices. Here the requirement for PvC environments is far more pretentious. It needs several large and small computing equipments and many plug&play [18], integrated or built-in I/O devices. Software is subclassified into the virtual machine [16], the window system and the graphical user interface, which must be properly designed under HCI guidelines [15]. The network aspect is subdivided into fixed Wired networks (*Ethernet LAN*, *Modem via dial-up telephone*), Wireless networks (*Wireless LAN*, *DECT*, *CDPD*, *GSM*) and Infrared (inside a room) with associated transfer protocols and rates [17]. Even satellite networks can be considered though the cost could be quite high [14]. However, architecture of a CSCW system under a PvC environment has to face many complications regarding data-transfer's traffic and delays, shared-data consistency, decomposition of large functionality to fit into small devices with limited capacity, etc.

CSCW architecture. CSCW software systems can be subdivided into four classes of features or devices: (1) input, (2) output, (3) application, and (4) data. Each of the four can be replicated or centralized. In a replicated application, each user's machine runs the application. An electronic multiuser whiteboard provides several pens as replicated input devices, but all other features can be centralized (see [10]). Centralizing the data guarantees consistency, despite high traffic to the application or I/O devices (see [12]). Replicated data requires less traffic for updates, but consistency is a more difficult issue.

3. Conclusions and Future Work

CSCW systems have become a place for proper communication on collaborative environments. According to the groupware typology different technologies have been used to give support to implement its corresponding system. Though basic technology for communication annoy people with too many computer-related tasks to deliver at-the-time elaborated notes in the meanwhile of a

group activity. An emerging trend such that ubiquitous, or pervasive computing promises to bring a solution on this matter. For this an overview of basic requirements for CSCW was presented and a typology of groupware options. They were analysed at the light of PvC systems by presenting current efforts that cover different aspects of CSCW requirements. As a groupware option a CSCW system under a PvC environments may be considered as a new and particular form of such systems; what can be called as PvCE-SCW systems.

Certainly there is much more to be discussed on these matters. Especially architectural aspects of every approach should be carefully studied and distilled such that proven solutions of PvCE-SCW requirements would be properly documented for usage in future developments.

4. References

1. Gruding J. *CSCW: History and Focus*. Computer Magazine, IEEE Computer Society. Vol 27, N.5, pg. 19-26. May 1994.
2. Mark W. *Turning Pervasive Computing Into Mediated Spaces*. IBM Systems Journal-Pervasive Computing Issue, Vol. 38, No. 4, *http://www.research.ibm.com/journal/sj38-4.html.* 1999.
3. Vichr R., Malhotra V. *Build it to move: The Architecture of Pervasive Computing*. IBM developerWorks: Wireless articles. May 2002. *http://www-106.ibm.com/developerworks/wireless/library/wi-archpvc/index.html.*
4. Wang Z., Garlan D. *Task-Driven Computing*. Carnegie Mellon University, School of Computer Science, Technical Report CMU-CS-00-154. May 2000. *http://reports-archive.adm.cs.cmu.edu/anon/2000/abstracts/00-154.html.*
5. Weiser M. *The Computer for the 21st Century*. Scientific American, 265(3): 94-104, September 1991. *http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html.*
6. Reinhard W., Shweitzer J., Völksen G. *CSCW Tools: Concepts and Architectures*. Computer Magazine, IEEE Computer Society. Vol 27, N.5, pg. 28-36. May 1994.
7. Tandler P. *Software Infrastructure for Ubiquitous Computing Environments: Supporting Synchronous Collaboration with Heterogeneous Devices*. In UbiComp'01. Heidelberg Springer LNCS 2201, pp.96-115, 2001.
8. Tandler P. *Modeling Groupware Supporting Synchronous Collaboration with Heterogeneous Single- and Multi-User Devices*. In CRIWG'01, IEEE CS Press, Darmstadt, Sep. 6–8, 2001.
9. Dorohonceanu B., Sletterink B., Marsic I. *A Novel User Interface for Group Collaboration*. In Proc. of Hawai Int'l Conference On System Sciences, IEEE. Maui, Hawai, January 2000.
10. Trimble J., Wales R., Gossweiler R. MERBoard. NASA Position Paper for the CSCW 2002 Workshop on Public, Community and Situated Displays.
11. Han H., Perret V., Naghshineh M. WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing. In Proc. of CSCW'00, ACM. Philadelphia, PA. December 2000.
12. Miranda H., Antunes M., Rodrigues L., Silva A. Group Communication Support for Dependable Multi-User Object Oriented Environments. 2000. Available at www.esw.inesc.pt/~ars/ps/dsmgc2000.pdf
13. Garlan D. *Pervasive Computing and the Future of CSCW Systems.* Proc. of CSCW'00, Philadelphia, Dec. 2000.
14. Chalmers D., Sloman M. A *Survey of QoS in Mobile Computing Environments.* IEEE Communications Surveys, *http://www.comsoc.org/pubs/surveys*, 2nd Quarter 1999.
15. Zeidler A. *User Interface Design for Ubiquitous Computing: W@PNotes, anExample*. Databases and Distributed Systems Research Group Wilhelminenstr, Darmstadt.
16. Kantor M,Redmiles D. *Creating an Infrastructure for Ubiquitous Awareness*. Inf.&Comp.Sc., Univ. California.
17. Seydim A. *An Overview of Transaction Models in Mobile Environments.* Department of Computer Science and Engineering, Southern Methodist University, Dallas.
18. Universal Plug & Play*, http://www.upnp.org*