Journal of Theoretical and Applied Information Technology
10th March 2013. Vol. 49 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MANAGEMENT OF CONCURRENCY MODEL IN COOPERATIVE GROUP WORK

[1]MESSAOUDI FAYCAL, [2]MOUSSAOUI MIMOUN

[1]Department Mathematics and Computer Science, Faculty of science, University Mohamed Premier

[2]Department Mathematics and Computer Science, Faculty of science, University Mohamed Premier

E-mail: [1]faycal.messaoudi@usmba.ac.ma, [2]m.moussaoui@ump.ma

ABSTRACT

This paper aims at presenting the problems relating to the management of cooperation sessions. We can by no means tackle such problems in an isolated way because they depend on types of application achieved during a session of cooperation, on the type and number of instances allowed in a given moment, and on the coordination; i.e. the types of dependence between the actions of participants in a session. The aim of this paper, after defining the scope of the computer supported cooperative work, is the identification of the specificities of the field and the different problems to solve when activating a CSCW environment: Computer Supported Cooperative Work. We shall focus on the inherent aspects and the repartition of an environment as regards the coherence of data, and the coherence of copies tolerance to deficiencies.

Keywords: *Group Work, Coordination, Cooperation, CSCW*

1. INTRODUCTION

The evolution of the computer networks during the last decades has made communication between computers faster. It has also allowed a number of machines distant data exchange. This gave rise to the field of research: CSCW (Computer Supported Cooperative Work). The CSCW appeared as a field of research in the mid-1980’s [BS89]. It is situated at a crossroad between Computer Sciences, Human Sciences and other. In fact the CSCW combines shared systems, networks, man-machine interfaces, artificial intelligence, sociology, psychology, and the organisation of work. Its objective is the study of multi-user interactive systems allowing several people to work together. The French terms collecticiel and synergiciel refer to such interactive systems. Anglo-Saxon terminology uses the term groupware to refer to a ‘collecticiel.’ The term ‘groupware’ was created by Peter and Trudy Johnson-Lens [1] in 1982 to refer to a set of information systems and the social processes of the groups supported by such systems. Several encyclopaedic dictionaries in Computer Sciences [2] define the ‘groupware’ on the basis of Ellis’ and all's definition [3]: «A computer-based system which supports groups of persons collectively carrying on a certain task or objective which provides an interface that gives access to a common environment». In accordance with [4] and within groupware systems, each user should be aware of the context in which his/her tasks are carried on. This implies the awareness of the existence of other users and their actions. In accordance with [5], the aim of CSCW is to examine how people work as a team towards a common objective, how groupware applications can improve communication between the members of the team at work within a CSCW environment, and how to guarantee coordination between them. According to [6] we can classify the different groupware types of applications according to two dimensions: space and time (moment); as such groupwares are systems that direct teamwork in four manners:

- Synchronic and coincident: same moment same place.
- Synchronic and displaced: same moment and different place.
- Asynchronic and coincident: different moment and same place.
- Asynchronic and displaced: different moment and different place.

Synchronous groupware is the class of applications in which two or more people collaborate in what they perceive to be real time. Examples of synchronous groupware are chat systems, shared whiteboards, group decision support systems, shared worlds, and collaborative editors. Greif [7] and Baecker [8] have useful background readings on such synchronous groupware.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Synchronous groupware is a natural complement to existing asynchronous groupware systems such as Lotus Notes. Asynchronous groupware can be seen as the electronic analog of work practices derived from the construction, circulation, and filing of documents; similarly, synchronous groupware can be seen as the electronic analog of work practices derived from meetings. But whereas asynchronous groupware is a well-established commercial market, synchronous groupware is still largely a research topic.

This article intends to introduce the problems relating to the management of the sessions of cooperation. We can by no means examine this kind of problems in an isolated way because they depend on the types of applications carried on at a session, , on the type and number of instances allowed in a given moment, and on the coordination; i.e. the types of dependence between the actions of participants in a session. The aim of this article, after defining the scope of the computer supported cooperative work, is the identification of the specificities of the field and the different problems to solve when activating a CSCW environment: Computer Supported Cooperative Work. We shall focus on the inherent aspects and the repartition of an environment as regards the coherence of data, and the coherence of copies tolerance to deficiencies. In this article, we provide the definition of the key concepts used in cooperative work. We shall then use these concepts to define the characteristics and specificities of synchronic groupwares and the functional needs they should meet. This article reflects our reflection on the groupwares through different research works conducted in the field of cooperative systems. Moreover, this study examines the link between the different taxonomies proposed in the domain of groupwares and aspects of architecture systems. Our objective in this study is to design a generic architecture that provides a number of functions which meet the identified needs. The use of these functions would allow us to:

- Put forward some cooperative tasks,
- Facilitate the construction of synchronic groupware,
- Reuse the same applications in order to achieve various scenarios of cooperation.

We wish to develop an architecture representing with codified solutions to commonly occurring problems. In this paper we have presented three classes of such solutions for the problem of developing groupware:

- Reference models,
- Architectural styles,
- Distribution architectures.

As the problems posed by these requirements become better understood, the architectural solutions presented in this article will necessarily be adapted, extended, and replaced.

Ultimately, we expect that support for groupware applications will be incorporated directly into mainstream operating systems, in much the same way that support for graphical applications has been gradually added over the past fifteen years. The current challenge is to determine the architectural abstractions and infrastructure that such support will require.

2. PROBLEMATIC OF COORDINATION WITHIN THE GROUPWARE SYSTEMS

In the groupware system, inevitable conflicts are certainly expected when several people work within the same. For instance two people might feel like moving the same objects in two different ways. In this case, there are two ways of sorting out the problem; either let the users sort it out between them or implement a mechanism in the application to solve it. In the first case, we talk about social protocol and in the second about material or technical protocol. The first solution is not necessarily a bad one. It has at least the advantage of being easy to implement. It is equally noted in [9] that it favours the users' development of their own protocols of the use of the groupware tool that is allotted to them. We can equally warn the user and let him/her solve the conflict. For instance in MPCAL (System of Management of calendar), if the constraints of meeting are incompatible, the users involved in the conflict are notified and have to solve it. In the case of concurrent access to the same document, the users are informed about it. They are expected either to ignore the warning taking into account that there is the risk and danger of the modification of the document [10]. We tend however to let the machine solve the maximum of conflicts possible. To automatically work out conflicts [11], several methods are possible:

- Locking
- Transactions
- The users take the floor each his/her turn.
- Centralised control
- Detection of dependencies
- Reversible execution
- Transformation of operations.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. GROUPWARE PARAMETERS

3.1. Modes of Interaction: Synchronic / Asynchronic Groupwares

Groupwares provide several possibilities to exchange information between participants. Groupware is said to be synchronic if information is exchanged according to a real mode. In this case, the actors are notified in real time of actions conducted on shared resources. The synchronic mode of interaction requires that the members of the group be present at the same time to carry on cooperative work. Tele-conference systems are representative examples of this mode of functioning [12]. This mode allows immediate sharing of information Conversely, asynchronic groupwares, correspond with cooperative applications whereby the support of communication is asynchronic and of an email type. The mode of asychronic interaction does not necessitate the co-presence of the participants. This mode is usually used in the domain of cooperative edition of documents. The asynchronic mode allows a sequential sharing of information. The classification of groupwares according to the mode interaction does completely recover all the cooperative tasks. Cooperative edition is an example of groupwares that can be both sychronic and asynchronic .

3.2. Granularity

Granularity is a spatial characteristic which is, unlike the mode of interaction, frequently associated with the rights of access to the data for the users of the groupware. Granularity defines the unit of information (the word, the paragraph, document, etc...) which can be accessed simultaneously by several users. A light granularity associated to a mode of synchronic interaction defines a work that is strongly coupled. In an opposite case, a strong granularity is associated to an asynchronic mode that is weakly coupled. An text editor, for example, who allows simultaneous access to a word allows a lighter granularity than one who does not allow simultaneous access to a document for another user In the case of groupwares who espouse the «token» technique, if the user is allowed access to a document, we are in front of a strong granularity.

3.3. Right of access, right of speech, «token» policy

The rights of access related to a user partly consist of attributions that are set to it thanks to its role, and partly by the rights that are accorded to it in the course of cooperative work. These dynamically attributed rights are commonly called the rights of speech. In the case of groupware, the right of speech is exchanged between the participants by dint of a «token» policy. Here it is a question of different protocols which allow participants to communicate control information by designating a «token» software. By and large, there is recourse to a single «token». This method assures exclusive access to shared information. Yet, in certain applications, several «tokens» can exist side by side, and this is often the case in virtual games where m users among n can modify the shared context. The strategies of passing the "token" are done in two ways. They are respectively based on:

- Explicit indication. An actor is charged to explicitly pass on the «token» to another member of the group. He/she can be the last possessor of the «token» or an actor who assumes the responsibility the intervention of the participants,
- Implicit handing of the «token». In this case, the actors don't intervene explicitly in the handing of the "token". It is the application that indicates the possessor of the «token» following the order of the users' request for the acquisition of the «token» (order FIFO), or also following a certain priority set up by the participants.

3.4. Free Access, Conflicts Of Access and Synchronization

The adoption of the mode free access can create conflicts and incoherence in the shared space. A simple example of such a situation which is notices when two conflicting operations (Operations of Writing): op. 1, op. 2 are carried on by two actors on the same object obj.1 (obj.1 is duplicated in all the sites of participants). If the execution of these two operations is not carried out following the same order the state of obj.1 will be incoherent. In order to guarantee coherence in synchronic groupwares, the system should synchronise the actions of all the participants in all the sites so as to keep a state of coherence on all data. GroupDesign [13] is a synchronic groupware of shared design. It provides free access mode to shared data. Two modes of functioning are provided by the system: strict WYSIWIS mode, and relaxed WYSIWIS mode. The relaxed is founded on a «validation» operation which aims at reflecting the actions achieved by the other participants in the other groupware sites. This mode of interaction appears to us to be rather asynchronic. As regards the functioning in strict WYSIWIS, GroupDesign allows free access to data which is similar to that used in Grove [14]. Group-Design provides a protocol for the solution of conflicts based on notion of logical clocks [15].

Journal of Theoretical and Applied Information Technology
10th March 2013. Vol. 49 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Each action conducted in the shared context leads to the creation and emission of an event stamped by the value of the local clock of the site. Each of the events encloses information such as: code of the executed operation, indicator of the object touched by the operation, and date of creation. The protocol maintains a coherent state of the ensemble of the shared data; it favours the execution of local actions. The protocol can, in certain configurations, remove any local execution if it proves out that some conflicting actions in other sites precede the execution of local actions.

4. DYNAMIC MANAGEMENT OF GROUPS: PROTOCOLS OF CONNECTION / DISCONNECTION OF PARTICIPANTS OF THE GROUPWARE

A service of the management of the group in a groupware basically provides protocols allowing: the connection of new members and disconnection of the old ones. Dynamic participation of participants is one of the functions provided by a synchronic groupware. This function marks the movement of a user from a private environment of work to a cooperative one. In order to ascertain the flexibility of use, the groupware should enable late comers to get connected in the course of work. The complexity of this operation consists in providing the new comer with a coherent view of the shared context, and notifying the members of the groupware of the logging in of new members. Moreover, the groupware should administer the departure of certain members whether voluntarily or due to network or site problems. To sum up, two constraints govern the execution of the functions of participation and connection. They are respectively spatial and temporal:

- Coherence of shared context: concerning the groupwares espousing the WYSIWIS mode of functioning at the opening phase of connection, the new-comer should have a similar view of the shared context. In this case, the protocol allows a strong coherence,
- Time of response: the process of the applications of connection or disconnection should be asynchronic. The protocol of connection should assure times of weak response for the applicant for connection as well as for the groupware actors.

5. CONCLUSION

In a cooperative teamwork, the members of the group exchange information in order to work together and in order to achieve different distant interactions on the space of production. The main aim of this work is to set up a formal frame for the coordination of a cooperative a distant work. To be able to reach this objective in an efficient way, it is compulsory to know first the manner in which information is exchanged between the members of the group. In fact, cooperation enables the participants to achieve different objectives that one single individual won't be able to reach. This means that the of the members of the group do not act in an isolated way, but rather interact with each other in the course of the execution, facilitating the attainment of the objective of cooperation. It is required then that the interactions respect a number of set rules of coordination. The coherence of data is evaluated at several levels, X. Warg et al. In [16] defines a model at three levels:

- Coherence at the level operations: Operations on the data should be carried on in a coherent order; for example, working on a shared object that has been deleted represents a coherence.
- Coherence at the level of syntax and structure of shared objects: The syntax and structure of shared objects should be the same for all users such as drawing in application of a shared blank chart.
- Coherence at the level of semantics relating to the meaning of data: Semantics relating to the meaning of data has to be common to all collaborating users. In addition to the problematic and of coherence and conflict in the course of operations, there is one more problematic caused by the internet, namely security policies and time of response. In fact, B. Broon, C. Sun et al. show in [17] that the time spent waiting for response becomes weighty for the users when the time of response is superior to 0.1 second. Coherence of context is not absolute; it is associated with the role, and even with the view allotted to the user as he/she joins the group. For instance, if a user is only allowed to share a particular application of the groupware in the WYSIWIS mode, the service manages the group provides him/her only with the context of application in question. The disconnection function marks the logging out of an actor of the groups of participants. The protocol of disconnection should guarantee a normal functioning of the groupware during this phase. In general, the protocol should make sure that the participant doesn't have a «token», or doesn't have a role which can disturb the evolution of the work (For example, the role of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the chair should be quitted before logging out). Dynamic connection/disconnection operations are carried out conforming to protocols that are often selected in accordance with the type of cooperative work. In certain groupwares, the connection of a user can be done through the consent of a chair. In other applications such as tele-conferences simulating for example a public seminar, the connection of a new participant is completely visible to the other actors of the groupware.

REFRENCES:

[1] Peter Johnson-Lenz, "Groupware: The Process and Impacts of Design Choices". In E. B. Kerr & S. R. Hiltz (eds), *Computer-Mediated Communication Systemes*: Status and Evaluation, Academic Press, N.Y., 1982, pp. 45-55.

[2] Anthony Ralston, Edwin D. Reilly et David Hemmendiger, Encyclopaedia of Computer Science, Nature Publishing Group, UK, 2000.

[3] Clarence A. Ellis, S. J. Gibbs, et G. L. Rein, "Groupware: Some Issues and Experiences", *Communications of the ACM* , janvier 1991, vol 34, Numero 1. TCAO, pp. 39-58.

[4] Anthony Ralston, Edwin D. Reilly et David Hemmendiger, "Encyclopedia of Computer Science" op.cit.

[5] G. Texier et N. Plouzeau, "Automatic Management of Sessions in Shared Spaces International" *Proceedings of the Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'99), Las Vegas (USA)*, 28 July – 1 Juillet , 1999, Vol. IV, pp. 2115-2121.

[6] T. Bray, J. Paoli, C. M. Sperberg –MCQueen, "Extensible Markup Language XML 1.0" *in W3C Recommandation* February 10, 1998.

[7] I. Greif (Ed) Computer-Supported Cooperative Work: *A Book of Readings. Morgan Kaufmann Publishers*, San Mateo, CA, 1988.

[8] R.M. Baecker (Ed) Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration. *Morgan Kaufmann Publishers, San Mateo*, CA 1993.

[9] C. De Koven and T. Radhakrishnan, "An Experiment in Distributed Group Problem Solving". *In Simon J. Gibbs and A.A. Verrijn-Stuart (eds)*, Multi-User Interfaces and Applications, pp. 61--76. IFIP WG 8.4, North-Holland, September 1990.

[10] Irene Greif and Sunil Sarin. Data Sharing in Groupwork. ACM Transactions on Office Information Systems, 5(2):187--211, April 1987.

[11] Clarence A. Ellis, S.J. Gibbs, et G.L. Rein, "Groupware: Some Issues and Experiences", *Communications of the ACM* , Janvier 1991, vol . 34, Numero 1. TCAO, pp. 39-58.

[12] T. Crowley, P. Milazzo, E. Baker, H. Forsdick, R. Tomlinson, "MMconf: An Infrastructure for Building Shared Multimedia Applications'', *CSCW'90*, pp. 329−342, Los Angeles, October 1990.

[13] A. Karsenty, M. Beaudouin−Lafon, "An Algorithm for Distributed Groupware Applications'', *In Proceedings of the 13 th Intl. Conf.* on Distributed Computing Systems ICDCS'93, pp. 195−202, IEEE, Pittsburgh state/USA, Mai 1993.

[14] C. A. Ellis, S. J. Gibbs, and G. L. Rein. "Groupware: Some Issues and Experiences." *Communications of the ACM*, vol. 34/1, pp. 39-58, Januray 1991.

[15] L. Lamport, ''Time, Clocks, and the Ordering of Events in a Distributed System'', *Communcations of the ACM*, 13(2), pp. 125−133, Juillet 1978.

[16] X .Wang, J. Bu, and C. Chen, "A new Consistency Model in Collaborative Editing Systems", *Fourth International ACM Workshop on collaborative Editing (CSCW'02)*, Louisiana, USA, 4 pages (CD Proceedings) November 2002.

[17] S. Sankaran and T. Bui. "A Web-based Correctional Telemedicine System with Distributed Expertise". *33 fd Hawaii International Conference on System Sciences, Island of Maui*, Vol. 5(5019). 10 pages (CD Proceedings), Hawaii – USA, January 2000.