Rethinking CSCW systems: the architecture of MILANO

Alessandra Agostini, Giorgio De Michelis
Cooperation Technologies Laboratory, DSI - University of Milano, Italy
{agostini, gdemich}@dsi.unimi.it

Maria Antonietta Grasso
Rank Xerox Research Centre, Grenoble Laboratory, France
Antonietta.Grasso@Grenoble.RXRC.Xerox.com

Abstract: After eleven years, CSCW is a well recognized research field which has generated, among other things, some new theoretical findings on work practices and cooperation and some new systems that are successfully applied by several organizations. The evaluation of successful applications from the point of view of the above recalled CSCW theories indicates some requirements (openness, continuity, contextualization and language-action integration) that the new generation of CSCW systems should satisfy. The prototype of the MILANO system is a working example of how those requirements can be met and of the challenges a full development of the CSCW potential poses to system designers and developers.

Introduction

Over the eleven years since the first CSCW Conference in Austin Texas, the CSCW field has grown and matured in America, Asia and Europe without having lost its primary characteristics: that of being an interdisciplinary research field involving people from both computer and human sciences.

Two facts have occurred during these eleven years that deserve attention:

1. while many failure cases have been reported and discussed in the literature (Grudin, 1988), some CSCW platforms, offering to its users a well equipped workspace, have been successfully applied and they are currently having a large diffusion in the market (the most relevant example is LOTUS NOTES,

J. Hughes et al (eds), Proceedings of the Fifth European Conference on Computer Supported Cooperative Work, 33-48.

followed by LINKWORKS and other Workflow Management Systems): we have therefore real feedback from users (Orlikowski, 1992; Bowers, 1994; Rogers, 1994; Rouncefield et al., 1994; Bowers et al., 1995; Ciborra, 1996; Prinz & Kolvenbach, 1996; Whittaker 1996);

2. it is becoming ever clearer to CSCW practitioners and designers that new CSCW systems should reflect the understanding of work practices developed over these years by anthropologists, organizational theorists and social scientists (Winograd & Flores, 1986; Swenson et al., 1994; De Michelis & Grasso, 1994; Bogia & Kaplan, 1995; Dourish et al., 1996).

In the process of developing new CSCW systems or prototypes, therefore, the crucial step is not testing a new cooperative application in some (semi)-real experimental work setting, but designing it so that it overcomes the limitations and drawbacks of existing CSCW platforms emerging from the analysis of their applications. From this point of view, it is crucial to analyze the existing platforms (in particular the successful ones, because their limitations are better distinguishable as the users do not express them through a radical refusal of the whole platform) in order to understand the new requirements to be met by the system to be designed.

In this paper we present the rationale and architecture of the new CSCW system prototype we have developed at the Cooperation Technologies Laboratory of the University of Milano, called exactly MILANO, in the context of a general discussion of the requirements a new generation of CSCW platforms should satisfy. Our point of view is rather general and can be considered our synthesis of the understanding we get through CSCW theoretical work (mainly the ethnographic and sociological research) of some successful applications of two CSCW systems - LOTUS NOTES, the only CSCW system with a massive worldwide diffusion (Ciborra, 1996; Whittaker, 1996), and LINKWORKS, whose application at a German Ministry is a surprising success (Prinz & Kolvenbach, 1996). Our choice has also been influenced by the fact that both systems are generic platforms (integrating various CSCW and traditional applications) for supporting work processes, like MILANO.

The theoretical perspective from which we observe work practices can be considered a situated language/action perspective, since it merges the situated action approach (Suchman, 1987; Lave & Wenger, 1991; Brown & Duguid, 1991) with the language/action perspective (Winograd & Flores, 1986). Work practices have the shape of cooperative processes, where people act and interact in order to do the required performance. Cooperative processes are performed by groups of people (communities) who share an experience of action, communication and learning. The community performing a cooperative process is a social aggregate constituted by all the people participating in it (both performers and customers); it is not a well defined organizational structure. The reader may look at (De Michelis, 1995, 1996, 1997) for further details on our theoretical approach to cooperative processes.

From the above perspective we synthesize the evaluations of some real cases of successful application of LOTUS NOTES reported in the literature (Ciborra, 1996;

Whittaker, 1996) and of the already mentioned application of LINKWORKS at a German Ministry (Prinz & Kolvenbach, 1996).

Based on the above analysis, we have defined four general requirements - openness, multimedia continuity, contextualization and integration of communication and action - new CSCW systems should meet as completely as possible, since their limited satisfaction in the above systems are responsible for both the success and limitations of the above cases. We have complemented them with a fifth requirement - personalized and selective workspace interfaces - that is not emerging from the above cases, but that we think will be increasingly more relevant as, and if, CSCW systems shape the workspaces of their users.

In this paper, after the discussion of the above requirements, we present the architecture of MILANO and its most relevant features. The conclusion is devoted to the discussion of some open problems and of our future research directions.

Some Requirements for a New Generation of CSCW Systems

As anticipated in the previous section, some general findings about CSCW systems, the analysis of some successful applications of LOTUS NOTES and of LINKWORKS and their conceptualization within the cooperative process theoretical perspective have led us to define four requirements for a new generation of CSCW systems. The presentation of the four requirements that follows is rather short and general: the references to the literature allow the reader to deepen her understanding of each.

Openness

Membership in a community is intrinsic and in many senses dynamic (Beck & Bellotti, 1993; Whittaker, 1996).

On the one hand, during the time interval within which a cooperative process is performed some members leave it while some new members join it. On the other, different members have different levels of engagement in it at different moments; finally, others participate in it occasionally. There is therefore a continuous movement between central and peripheral participation (Lave & Wenger, 1991) in the cooperative process.

The CSCW system supporting a cooperative process should adapt itself to current membership with the maximum degree of plasticity in order to avoid constraining the behavior of the community members. In order to do so, it should first and foremost be able to support the interaction between those having it and those not having it. LOTUS NOTES is going in this direction with its INTERNOTES module.

Multimedia Continuity

Communities performing a cooperative process are distributed in space and time. Their members communicate through different media not only because at any moment one of them can use only some media, but also because she needs to choose the best available medium for the communication she needs to do. They need therefore to be supported by a system not only making available a large set of communication media, but also allowing them to switch almost continuously from one medium to another so that they can choose at any time and in any situation the best available medium (Reder & Schwab, 1990; Whittaker, 1996; G. Patriotta in (Ciborra, 1996)). The application of LINKWORKS at the German Ministry has widely extended it in this direction (Prinz & Kolvenbach, 1996).

Contextualization

A cooperative process is a history of mutually related communication and action events. Any communication (action) event logically follows some communication and/or action events, except for the event starting a new cooperative process; and it triggers other communication and/or action events. The ordering among the events of a cooperative process therefore selects within the temporal evolution of the events the links defining a dependence relation. Thus the context of a cooperative process is mirrored by the partial order of communication and action events representing its history. The actors of a cooperative process are immersed in its history and need to refer to a representation of it in order to act effectively (Orlikowski, 1992; Prinz & Kolvenbach, 1996; Whittaker, 1996; W. Orlikowski in (Ciborra, 1996)). Sometimes they need the context to be transparent, other times visible (Agostini et al., 1996). LOTUS NOTES supports partially contextualization through the possibility of attaching a conversation to a document, while LINKWORKS offers a greater assistance with its capability to create a folder for any cooperative process where the documents created during its execution are stored and mutually linked.

Integration between Communication Flow and Action Flow

Within a cooperative process people communicate and act. Both communication and action flows define the basic units of cooperative work: respectively, conversations and workflows.

On the one hand, within the communication flow of a cooperative process we can observe various distinct conversations; i.e., various partial orders of mutually related communication events (Bullen & Bennett, 1990). On the other, within the action flow of a cooperative process we can distinguish various types of structures of mutually related action events:

- partial orders of actions performed in agreement with a plan designed outside the process (let us call it a procedure);
- partial orders of actions performed in agreement with a plan designed inside the process (project);
- partial orders of actions created step by step (evolutionary workflow).

Plans for both projects and procedures are resources for actions (Suchman, 1987): they must be simple, open to exceptional paths and/or to changes (Abbott & Sarin, 1994; Swenson et al. 1994, Dourish et al., 1996; Ellis et al., 1995).

Communication events and action events (and, therefore, conversations and workflows) are mutually related since actions are agreed, delivered and declared (un)successful within conversations, whereas conversations can be generated (as a reaction to a breakdown) while performing an activity. A CSCW system supporting a cooperative process should provide its users a support to both conversations and workflows and to their integration. Both LOTUS NOTES and LINKWORKS are partially meeting the above requirement since they allow users to create conversations within workflows but not vice versa.

Personalized and Selective Workspace

Many intensive users of workstations experience problems managing the objects they have created, received and/or manipulated in their workspaces. In order to give order to them, they continuously create new folders, folders of folders, and so on. Names of objects and/or folders become important for retrieving them; after a while previous names of folders and/or objects lose meaning, and finding things needed becomes increasingly more difficult. The situation becomes more difficult if and when the workspace is populated also by objects the user shares with other people - as in the case of CSCW systems - giving the user the impression that the situation is beyond her control. The user has, in fact, to manage not only the objects she has located somewhere but also the objects created by other users and located by the system automatically, so that she is no longer aware of her workspace. In our opinion, this fact will create a major obstacle to the diffusion and use of CSCW systems. A CSCW system supporting a community should, at any moment, bring forth to any of its users all and only the objects she needs at that particular moment. A reasonable criterion for choosing what to display at a particular moment to a particular user is to relate the distance of an object in the workspace (Rodden, 1996) to the time distance of the cooperative process to whose context it belongs from the current activity performed by the user.

This requirement is not sufficient *per se*, since the context of a cooperative process is a complex issue. The actors of a cooperative process are, in fact, immersed in a unique history. But due to their different roles in the process they have different views of that history; i.e., of the past events of the process in which they are participating (Malone et al., 1992; Trigg & Bødker, 1994). Moreover, within her personal view of the past events of the process, each participant at any time is looking at a part of it, depending on what she is currently doing. The personal view of each member of a community mirrors the context in which she is acting and interacting. A CSCW system supporting a cooperative process should provide all its users with the right context for their participation, increasing their awareness of it. Users sometimes need the context to be transparent, other times visible (Agostini et al., 1996).

Shaping the workspace offered by a CSCW system through the capability of providing its users with personalized views and automatic filtering mechanisms is a rather difficult task. The workspace becomes in some sense alive, since it changes automatically reacting to what its user is doing and to the communication of the other users. There is, however, the risk that it confuses the user as much as traditional interfaces do, even if in a different way. The problem is therefore to design the workspace in a 'natural' way, so that its behavior does not capture the attention of the user who can act transparently in it. This is a typical 'design' problem in the sense described by Terry Winograd (1996).

While the MILANO system (as will be shown in the next section) has been conceived to meet to a certain degree the four previous requirements, research in regard to the design of its workspace is still at an exploratory stage. New solutions are under development in cooperation with Marco Susani of Domus Academy.

Let us conclude this section summarizing in Table I at which degree both LOTUS NOTES and LINKWORKS achieve the above listed requirements. We claim that the successful LOTUS NOTES cases rest on its approach to the openness issue and on its capability to link the communication events to the outcomes of the actions (even if the two flows are not completely integrated). In reference to LINKWORKS, as used and extended in the Politeam project (Prinz & Kolvenbach, 1996), we believe that its strong points are based on devoting considerable care to the multimedia issue, as well as on providing their users with the historical context of the process; moreover, the integration of the action flow with the documents related to communication events allows a medium level of integration between communication and action.

Requirements	Low	Medium	High
Openness	LINKWORKS		LOTUS NOTES
Multimedia Continuity	LOTUS NOTES	LINKWORKS	
Contextualization		LOTUS NOTES	LINKWORKS
Integration between Conversation and Action Flows		LOTUS NOTES, LINKWORKS	
Personalized and Selective Workspace	LOTUS NOTES, LINKWORKS		

Table I. Comparative evaluation of LOTUS NOTES and LINKWORKS

The Architecture of the MILANO System

The currently developed MILANO system prototype is a CSCW platform integrating in the user workspace a workflow management system (MWMS) with a multimedia conversation handler (MCH), and, finally, an object repository (MOR).

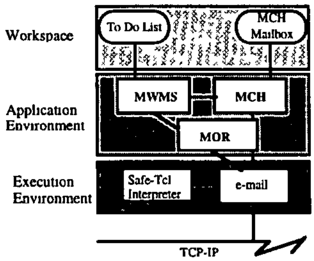

Figure 1. The overall architecture of the MILANO system

MILANO, evolving from previous mail-based CSCW systems (THE COORDINATOR (Winograd & Flores, 1986), INFORMATION LENS (Malone et al., 1986), CHAOS (De Cindio et al., 1986), STRUDEL (Shepherd et al., 1990), UTUCS (Agostini et al., 1994), CONVERSATION BUILDER (Bogia & Kaplan, 1995)), is based on the computational mail (through the Safe-Tcl environment see the next section). Future developments will be: on the one hand, the extension of the existing components and, in particular, a newly conceived workspace interface; on the other, the development of the links integrating MILANO with the organization information system (De Michelis et al., 1996).

The openness of the MILANO system architecture

As pointed out in the previous section, CSCW platforms need to be open with respect to both the integration of any software application and the cooperation with other systems. The latter requirement in particular finds its natural environment in the Internet. Openness of cooperative processes on the Net, we think, can be reached at a maximal degree through the enabled mail model. Enabled mail is an active medium, where messages are automatically manipulated when they are sent and/or received. E-mail is often recognized as one of the most successful groupware applications (after fax and telephone). At a functional level, it matches the way people work; it is asynchronous, easy to learn, and ready to be electronically stored. At a technical level, it runs on heterogeneous environments from its very beginning. Moreover, the use of e-mail as a distribution model makes the group boundaries inherently *dynamic*, providing contemporaneously different degrees of participation. E-mail is, therefore, a natural candidate to be the middleware for groupware, that is, its enabling technology. This capability increases significantly with enabled mail. An example of enabled mail is BEYOND

MAIL. Based on the INFORMATION LENS (Malone et al., 1986), it basically adds a rule engine to the mail agent, in such a way that users can define rules to be applied to the messages and to the documents routed in the system. Within the enabled mail family the term computational mail is used when the enabling of e-mail comes specifically from the embedding of programs within electronic mail messages (Borenstein, 1992). With computational mail, for example, messages can bring the execution environment necessary for reacting to them, providing the receiver with whatever she needs to be effective in the cooperation.

MILANO uses the computational mail model to obtain openness. In particular, the Safe-Tcl environment (Borenstein, 1994) has been used. Let us discuss this. The Safe-Tcl interpreter is based on an existing language called Tcl (Ousterhout, 1993) and on its graphical extension Tk. Safe-Tcl enriches the original Tcl primitives adding to them some commands to handle MIME messages; it also inherits the Tk primitives to design portable interfaces. Moreover, the security issue has been carefully considered designing a particular mechanism of safe interpretation. Finally, a mechanism of safe extendibility has been designed in such a way that the system can be aware of 'safe sites' on the net where it can look for extension scripts not found locally; the interpreting mechanism allows their use in 'real-time', avoiding the need to recompile the entire system.

MILANO makes great use of active software objects; i.e., self-contained, concurrently executing software processes, encapsulating some states and being able to communicate with other agents via message passing (Wooldridge & Jennings, 1995).

Within MILANO we distinguish three different kinds of active software objects: mail-robots, agents and applets. *Mail-robots* are specialized portions of code which are executed, without directly involving the user, either at delivery time or at receipt time. They are devoted to filtering the mail, to synchronizing information, and so on. In turn, *agents* are multipurpose and event-driven portions of code devoted to handling the objects of the system and their relationships. Both agents and mail-robots have full access to the user's resources. Finally, the *applets* - their name refers to Java applets, even if Safe-Tcl allows MILANO's applets to have a much broader set of functionality (Wayner, 1996) - are mobile and support interaction between MILANO users and those without the system installed at their site. An applet is a portion of code that can be inserted in a message and is executed at *activation* time, when the user reads it.

The above features allow MILANO to support, with different levels of service, cooperation between three types of actors: those having the MILANO system; those having the Safe-Tcl interpreter; and those just using e-mail (see Figure 2).

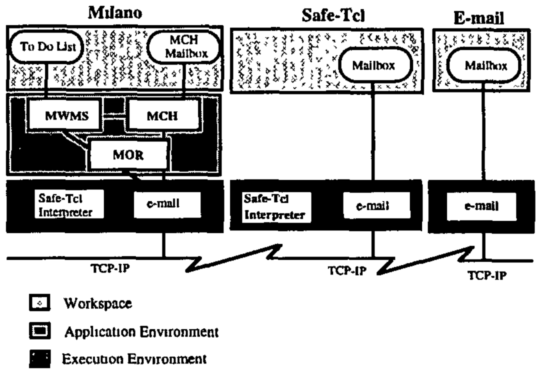

Figure 2. The multi-level user architecture of the MILANO system

While two actors both having MILANO can obviously cooperate using the full set of services offered by it; the other cases require further discussion. The actor having a Safe-Tcl interpreter is still provided with a meaningful subset of functionalities, since MILANO's messages embed data and source code that the interpreter executes on destination sites for interacting with MILANO users. Some of the functionalities provided are: notification of activities to be performed; visualization of the context of an activity; complete handling of the action flow (calculation of the next step, routing of the involved data, etc.); access to both the missing parts of a conversation and, more generally, to the state of the running cooperative process. This is done by the applet (see above) issuing back a request on the sender machine to the agent who is in charge of handling the conversations or the cooperative processes, as appropriate.

For her part, the actor who just uses common e-mail is also provided with useful information and can easily be involved in an ongoing cooperative process. In this case, in fact, any program embedded in a message cannot be executed so the user can only read it. This is why messages always contain a 'textual' part giving all relevant information, such as the description of the activity to be executed, the information to be produced and so on. In case of a user having a MIME compliant reader the 'textual' part of messages is actually hypermedial. Moreover, in order to handle the flow of information, messages embed forms for filling in some information to be returned to the sender, so that MILANO does not see any difference with respect to the other two cases.

The MILANO Object Repository (MOR)

The MILANO system reflects and represents the history of a cooperative process linking any new (communication or action) event to the events to which it is reacting: a reply is linked to the message to which it is replying; an action is linked either to the communication event where its performance was agreed upon or to the previous action within a plan; etc. Through these links the system automatically creates for any cooperative process a partial order, selectively offering the user access to the appropriate resources, tools, information and documents she needs to act effectively.

Figure 3 offers some hints about the representation of the history of a cooperative process in the MILANO system. The context presentation of its interface is currently under development along similar lines.

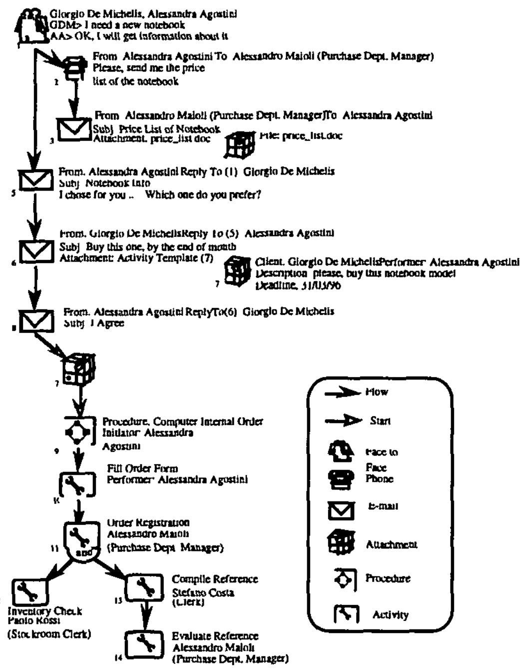

Figure 3. A sketch of the history of a cooperative process

The partial order of Figure 3 contains both communication and action events. The links relating them are distinguished between those connecting objects of the same type (Flow arrows) and those connecting objects of different type (Start arrows). Flow arrows single out from the history of the cooperative process the representation of conversations and plan instances, while Start arrows represent the integration between conversations and workflows.

Finally, it has to be recalled that MOR is a fully distributed component, based on both the user workstations and on the Web server of the whole community: the former contain the personal archives of each user, while the latter contains the object base where each user can search for the objects she does not have in her archive (see the subsection on Workspace below).

The MILANO Conversation Handler (MCH)

Conversations are the basic unit of communication in MILANO (De Michelis & Grasso, 1994): any communication event (message, meeting session, phone call, etc.) is always part of a conversation, within which it gets its sense. Milano conversations are unstructured, multipersons, multimedia; and allow any type of attachment.

A conversation acts as a folder collecting communication events from a group of participants which can been modified as needed. The communication events can be created by a complete range of communication media: from synchronous to asynchronous, supporting distant or close communication (only a subset of these possibilities is currently implemented in MILANO, but extensions are foreseen). Each event is characterized by the possibility of attaching enclosures of various types.

The MILANO Workflow Management System (MWMS)

As observed in the previous section, workflows may be either plan instances (procedures or projects) or evolutionary workflows. With respect to plans, the MWMS allows its users to design them even if they have little or no experience with computer science, programming, formal languages, etc., providing them with a design framework for defining simple plan models: a user can in fact design the plan as a partial order of progressing activities (steps), disregarding at the design phase any exception and/or breakdown which may occur during its execution. Given a plan definition, the set of possible exceptional paths is computed when needed. In general the exceptional paths allow the user to roll back, to jump forward, to execute unforeseen activities, etc. Moreover, exceptional paths are distinguished on the basis of the different levels of responsibility that can authorize them.

The workflow management component of the MILANO system embeds a part of the theory of Elementary Net Systems (Rozenberg & Thiagarajan, 1986; Nielsen et al., 1992) allowing an elegant treatment of both static and dynamic changes (Ellis et al., 1995). Therefore, the MWMS offers a framework for effectively supporting both procedures and projects.

Finally, the MILANO workflow module supports evolutionary workflows through its conversation handler (see next subsection).

Linkages between MWMS and MCH

Conversations and plans are not the only substructures of the partial order representing the history of a cooperative process: the latter also contains linkages connecting communication and action events (and, therefore, conversations and plans). These linkages between heterogeneous objects reflect their dependency relations: sometimes a conversation is subordinated to a plan; sometimes a plan is subordinated to a conversation.

On the one hand, besides electronic documents such as spreadsheets and graphical pictures, the activity template is a special kind of enclosure that can be created in any conversation when the need arises to formalize an action to be performed. When its requester and performer agree on its content, the activity template becomes an object of the to do list (see the next subsection) of both: that is, regarding the execution aspects, it is handled by the MWMS. This feature of MILANO allows its users to agree on various subsequent activity templates within a conversation, relating them to one another. In this way, the idea of electronic circulation folders proposed in (Karbe et al., 1990) and successfully employed in (Prinz & Kolvenbach, 1996) is realized to support evolutionary workflows as well as any combination of them with plans and/or procedures.

On the other, the breakdowns occurring during the execution of a project and/or of a procedure are discussed within a conversation, eventually generating an agreement on an activity template to perform an authorized exceptional path (see previous section). Figure 4 summarizes how plans, conversations, cooperative processes, activity templates are mutually related.

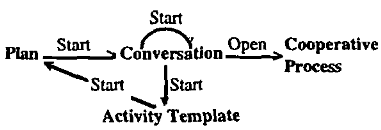

Figure 4. The relations among MILANO objects

The MILANO Workspace

As we claimed above, the context of a cooperative process is mirrored by the partial order of communication and action events representing its history; the participants of a cooperative process are immersed in this history and need to refer to a representation of it in order to act effectively.

MILANO supports tailorable personal views (Malone et al., 1992; Trigg & Bødker, 1994) of the history of a cooperative process with some original features: for instance it avoids information overload selecting automatically from the historical context only those events in which the user is involved, in order to avoid information overload (for example it hides from her all conversations in which she neither is the sender/receiver nor belongs to the carbon copy list); whereas it allows the initiator of a plan instance to see its whole history in her personal view, it makes visible to the other participants only those activities in which they are performers or clients. In fact whereas those objects (conversations, activities, plans, etc.) whose a person is a direct user are recorded in her local workstation, the others are recorded in the community server in the Web. Finally, a user can always modify her view when she wants to do it: she can choose which objects to see in a particular moment (by the filtering mechanisms on the context objects) and in which way to see them (by choosing which information is relevant for her and in what order).

Besides making visible to its users the context of a cooperative process, the MILANO system also supports two operational views allowing them to be active in it: the mailbox and the to do list (see Figures 5 and 6), corresponding respectively to communicating and acting.

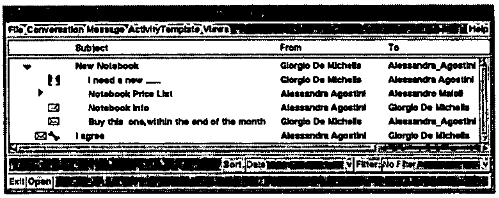

Figure 5. The mailbox presentation of the MILANO system

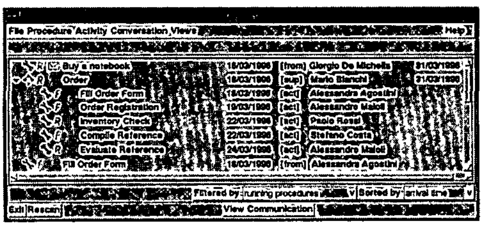

Figure 6. The to do list presentation of the MILANO system

To maintain multiple views of a single history MILANO must handle, as asserted in the MOR subsection, both the personal archives of each member on their workstations and the common archive of the whole community on a Web server.

Conclusion

Summarizing, the main innovative features of the MILANO system allowing it to satisfy the requirements proposed in the Introduction are the following:

- being based on the computational mail model, MILANO has an high degree of openness;
- since its conversations are unstructured, multipersons and multimedia, MILANO exhibits an high degree of continuity;
- since its conversations and workflows are mutually linked, MILANO integrates in a flexible and rich way the communication and action flows, offering a variety of representations of the history of a cooperative process.

It has to be underlined concluding this paper that the innovation proposed by MILANO is mainly architectural: with respect to some issues, in fact, it evolves from previous systems simplifying some features (conversations are *not* structured and *not* tailorizable; plans are *not* cyclic; etc.), while, with respect to other, it introduces new objects and mechanisms (conversations are multimedia; plans are based on a theoretical model allowing to compute run-time the exceptional paths; the users are always immersed in the context of the cooperative process in which they participate; etc.).

Acknowledgements

This paper presents a research that has been conducted with the financial support of the EC and of the Italian National Research Council (CNR). Special thanks to the anonymous referees of ECSCW'97 and of a previous conference, whose useful comments allowed us to improve the readability of the paper.

References

Abbott, K. R. and Sarin, S. K (1994): Experiences with Workflow Management Issues for the Next Generation. In *Proceedings of the Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, October 22-26 ACM Press, New York, pp 113-120.

Agostini, A., De Michelis, G., Patriarca, S. and Tinini, R (1994): A Prototype of an Integrated Coordination Support System. *Computer Supported Cooperative Work. An International Journal*, vol. 2, no. 4, pp 209-238.

Agostini, A., De Michelis, G., Grasso, M. A., Prinz, W. and Syri, A. (1996) Contexts, Work Processes and Workspaces *Computer Supported Cooperative Work. The Journal of Collaborative Computing*, vol. 5, no. 2-3, pp. 223-250.

Beck, E and Bellotti, V. M E (1993): Informed Opportunism as Strategy: Supporting Coordination in Distributed Collaborative Writing In *Proceedings of the Third European Conference on Computer Supported Cooperative Work*, Milano, Italy, September 13-17. Kluwer Academic, Dordrecht, pp. 233-248.

Bentley, R., Horstmann, T., Sikkel, K and Trevor, J (1997): The World Wide Web as Enabling Technology for CSCW: The case of BSCW. In *Computer Supported Cooperative Work. The Journal of Collaborative Computing*, (to appear).

Bogia, D P and Kaplan, S. (1995): Flexibility and Control for Dynamic Workflows in the wOrlds Environment. In *Proceedings of the Conference on Organizational Computing Systems*, Milpitas, CA, August 13-16. ACM Press, New York, pp. 148-159.

Borenstein, N. S. (1992). Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work. In *Proceedings of the Conference on Computer-Supported Cooperative Work*, Toronto, Canada, October 31-November 4. ACM Press, New York, pp. 67-74.

Borenstein, N. S. (1994). Email with a Mind of Its Own: The Safe-Tcl Language for Enabled Mail In *Proceedings of the IFIP WG 6.5 Conference on Upper Layer Protocols, Applications, and Architectures*, Barcelona, Spain, June.

Bowers, J. (1994) The Work to Make a Network Work: Studying CSCW in Action In *Proceedings of the Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, October 22-26. ACM Press, New York, pp. 287-298.

Bowers, J., Button, G. and Sharrock, W (1995): Workflow from Within and Without: Technology and Cooperative Work on the Print Industry Shopfloor In *Proceedings of the Fourth European Conference on Computer Supported Cooperative Work*, Stockholm, Sweden, September 10-14. Kluwer Academic, Dordrecht, pp 51-66.

Brown, J S and Duguid, P. (1991): Organizational Learning and Communities of Practice. a unified View of Working, Learning and Innovation. *Organization Science*, vol. 2, no. 1, pp. 40-56.

Bullen, C. V. and Bennett, J. L. (1990): Learning from User Experience with Groupware. In *Proceedings of the Conference on Computer-Supported Cooperative Work*, Los Angeles, CA, October 7-10 ACM Press, New York, pp. 291-302

Ciborra, C. U. (edited by) (1996): *Groupware & Teamwork. Invisible Aid or Technical Hindrance.* John Wiley and Sons Ltd., Chichester.

De Cindio, F., De Michelis, G , Simone, C., Vassallo, R. and Zanaboni, A. M. (1986): CHAOS as a Coordination Technology. In *Proceedings of the Conference on Computer Supported Cooperative Work* , Austin, TX, December 3-5. MCC, Austin, pp 325-342.

De Michelis, G (1995): Computer Support for Cooperative Work: Computers between Users and Social Complexity. In C. Zucchermaglio, S Bagnara and S. Stucky (eds.): *Organizational Learning and Technological Change.* Springer Verlag, Berlin, pp. 307-330.

De Michelis, G (1997). Cooperation and Knowledge Creation. In I. Nonaka and T. Nishiguchi (eds.). *Comparative Study of Knowledge Creation* Oxford University Press, Oxford (to appear).

De Michelis, G., Dubois, E , Jarke, M., Matthes, F , Mylopoulos, J., Pohl, K., Schmidt, J., Woo, C and Yu, E. (1996). Cooperative Information Systems: A Manifesto. In M. Papazoglou and G. Schlageter (eds.): *Cooperative Information Systems*, Academic Press (to appear).

De Michelis, G. and Grasso, M. A. (1994): Situating conversations within the language/action perspective: the Milan Conversation Model. In *Proceedings of the Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, October 22-26. ACM Press, New York, pp. 89-100.

Dourish, P., Holmes, J., MacLean, A., Marqvardsen, P and Zbyslaw, A. (1996): Freeflow: Mediating Between Representation and Action in Workflow Systems. In *Proceedings of the Conference on Computer Supported Cooperative Work*, Boston, MA, November 16-20. ACM Press, New York, pp. 190-198.

Ellis, C. A., Keddara, K. and Rozenberg, G. (1995): Dynamic Change within Workflow Systems. In *Proceedings of the Conference on Organizational Computing Systems*, Milpitas, CA, August 13-16. ACM Press, New York, pp. 10-21.

Grudin, J. (1988): Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces. In *Proceedings of the Conference on Computer Supported Cooperative Work*, Portland, OR, September 26-28 ACM, New York, pp. 85- 93.

Karbe, B., Ramsperger, N. and Weiss, P. (1990). Support of Cooperative Work by Electronic Circulation Folders. *ACM SIGOIS Bulletin*, vol 11, no 2-3, pp. 109-117

Lave, J and Wenger, E. (1991): *Situated learning. Legitimate peripheral participation* Cambridge University Press, Cambridge.

Malone, T. W., Grant, K R and Turbak, F.A. (1986): The Information Lens: An Intelligent System for Information Sharing in Organizations. In *Proceedings of the Conference on Human Factors in Computing Systems*, Boston, MA, April. ACM, New York

Malone, T. W., Lai, K.-Y and Fry, C. (1992): Experiments with Oval: A Radically Tailorable Tool for Cooperative Work. In *Proceedings of the Conference on Computer-Supported Cooperative Work,* Toronto, Canada, October 31 - November 4. ACM Press, New York, pp. 289-297.

Nielsen, M., Rozenberg, G. and Thiagarajan, P. S. (1992): Elementary Transition Systems. *Theoretical Computer Science*, vol. 96, no. 1, pp. 3-33.

Orlikowski, W. J. (1992): Learning from Notes: Organizational Issues in Groupware Implementation In *Proceedings of Conference on Computer Supported Cooperative Work,* Toronto, Canada, October 31 - November 4. ACM Press, New York, pp. 362-369

Ousterhout, J. K. (1993): *TCL and the TK Toolkit* Reading, Addison Wesley, MA.

Prinz, W. and Kolvenbach, S. (1996): Support for Workflows in a Ministerial Environment. In *Proceedings of the Conference on Computer Supported Cooperative Work,* Boston, MA, November 16-20. ACM Press, New York, pp. 199-208.

Reder, S. and Schwab, R. G. (1990): The Temporal Structure of Cooperative Activity. In *Proceedings of the Conference on Computer-Supported Cooperative Work,* Los Angeles, CA, October 7-10. ACM Press, New York, pp. 303-316.

Rodden, T. (1996) Populating the Application: A Model of Awareness for Cooperative Applications. In *Proceedings of the Conference on Computer Supported Cooperative Work,* Boston, MA, November 16-20. ACM Press, New York, pp. 87-96.

Rouncefield, M., Hughes, J. A., Rodden, T. and Viller, S. (1994): Working with "Constant Interruption". CSCW and the Small Office. In *Proceedings of the Conference on Computer Supported Cooperative Work,* Chapel Hill, NC, October 22-26. ACM Press, New York, pp. 275-286

Shepherd, A., Mayer, N. and Kuchinsky, A. (1990): Strudel - An Extensible Electronic Conversation Toolkit. In *Proceedings of the Conference on Computer Supported Cooperative Work,* Los Angeles, CA, October 7-10. ACM Press, New York, pp. 93-104.

Suchman, L. A. (1987): *Plans and Situated Actions. The problem of human-machine communication.* Cambridge University Press, Cambridge.

Swenson, K D., Maxwell, R J , Matsumoto, T., Saghari, B. and Irwin, K. (1994) A Business Process Environment Supporting Collaborative Planning. *Collaborative Computing*, vol. 1, no. 1, pp. 15-34.

Trigg, R. H. and Bødker, S. (1994): From Implementation to design: Tailoring and the Emergence of Systematization in CSCW. In *Proceedings of the Conference on Computer Supported Cooperative Work,* Chapel Hill, NC, October 22-26. ACM Press, New York, pp. 45-54.

Wayner, P. (1996): Net Programming for the Masses. *Byte,* vol. 21, no. 2, pp. 101-104.

Whittaker, S. (1996): Talking to Strangers. An Evaluation of the Factors Affecting Electronic Collaboration. In *Proceedings of the Conference on Computer Supported Cooperative Work,* Boston, MA, November 16-20. ACM Press, New York, pp. 409-418.

Winograd, T (edited by) (1996). *Bringing Design to Software.* Addison Wesley, New York.

Winograd, T. and Flores, F. (1986): *Understanding Computer and Cognition: A New Foundation for Design.* Ablex Publishing Corp., Norwood

Wooldridge, M. and Jennings, N R. (1995): Intelligents agents: theory and practice. *The Knowledge Engineering Review,* vol. 10, no. 2, pp. 115-152.