Human-Computer Interaction - INTERACT'99
Angela Sasse and Chriss Johnson (Editors)
Published by IOS Press, @ IFIP TC.13, 1999
(Please do not distribute)

How can groupware preserve our coordination skills? Designing for direct collaboration

Stéphane Sire, Stéphane Chatty,
Hélène Gaspard-Boulinc, François-Régis Colin

Centre d'Études de la Navigation Aérienne
7, avenue Edouard Belin
31055 Toulouse cedex
{sire, chatty, helene, fcolin}@cena.fr

ABSTRACT — Groupware systems often propose coordination protocols inspired from computer technologies. Such protocols are rigid compared to the subtle coordination hints and to the social rules used by humans. Protocols act as intermediate between users in the same way as command languages once did between users and tasks. We propose to reduce the role of those intermediates in what we call direct collaboration interfaces. We then explore design rules that support direct collaboration: media and activity integration, and interaction styles that support prosody and social hints. We finally describe an application to air traffic control.

KEYWORDS — Groupware, collaborative work, interaction style, air traffic control, direct collaboration

1. INTRODUCTION

Although research in the field has been very productive in the last ten years, there are still few applications of groupware systems in domains other than office systems. Among them, air traffic control (ATC) is probably one of the domains that would benefit most from advances in Computer Supported Collaborative Work (CSCW). A candidate application for groupware is the negotiation that occurs when a controller wants to modify the route of an aircraft in a way that affects another controller who works in another room, hundreds of kilometers away. Today's negotiations, carried through a telephone, routinely take half a minute for exchanges that would take less than ten seconds if the controllers were sitting next to each other. It is tempting to believe that adequate groupware support would help sparing those precious twenty seconds, and improve the comfort of negotiations.

However, providing the adequate groupware support is not easy. A classical technique used by engineers consists in analyzing telephone conversations between controllers, classifying the different types of exchanges and then proposing a computer equivalent for each of them. Taking advantage of the evolution of ATC systems towards graphical interactive systems (Chatty & Lecoanet, 1996), it is easy to extend data input with collaboration capacities. For instance, a controller who wants to allow an aircraft to climb can input the proposed flight level, then make a dialogue box appear on a colleague's screen, as shown in figure1.

Using the computer as an intermediate like this is technically easy, and looks like an efficient reuse of the user previous input actions. But is it efficient for negotiating with other users? This is far from obvious. As far as ATC is concerned, dialogue boxes like the one shown in figure 1 were not considered as fully satisfactory, because air traffic controllers felt that communications would be too constrained. We

consider this as a hint that there are mechanisms in human communication that are not easily captured by computer-mediated interaction.

Figure 1: Proposing a new flight level to another air traffic controller: a rigid protocol

In this article, we analyse how computer-based coordination protocols come in the way of collaboration and we introduce the notion of direct collaboration, by comparison with direct manipulation. We then propose some rules for designing direct collaboration systems, and we describe metaphors for data exchange that support direct collaboration. Finally, we take as an example the application of direct collaboration techniques to air traffic control.

2. COLLABORATION WITH CSCW

Collaboration with CSCW tools requires more than one application program or technology because each one often covers a limited set of requirements. Some of these applications impose coordination protocols on users.

2.1. Classification of Groupware Systems

The Clover model, a functional classification of groupware systems, decomposes collaborative work into three types of activities: **communication**, **production** and **coordination** (Calvary et al., 1997). Each of these activities is supported by different components of groupware systems. Communication is the activity aimed at exchanging information among persons. Production is the central activity of shared editors, it is composed of all the actions that aim at building a common artifact. Coordination is the activity that deals with resource allocation and dependencies between users' tasks.

Using the Clover model, a distinction between three types of groupware systems appears. *Media-spaces*, video and telephone applications are centered around communication, provide little coordination support and have no production component. *Shared editors* are centered around production, have a strong coordination component, and provide little communication. Finally, a few systems integrate shared data editing and communication channels (Buxton, 1992). Their communication and production components are both strong, while their coordination component is variable. For example, Clearboard merges a video connection and a shared drawing tool by superposing them on the same screen with a transparent overlay technique (Ishii & Kobayashi, 1992).

2.2. Coordination Protocols

When people produce documents or other artifacts in common, they coordinate their actions with each other. They apply coordination **protocols** for that purpose. Anyway, the introduction of groupware systems modifies the nature of these protocols. Most shared editors provide functions that impose new coordination protocols on users. For instance turn-taking is a simple protocol that prevents two persons from modifying the same piece of data at the same time. Unlike their real-world equivalent, when users are next to each other and do not need groupware assistance, these mechanisms are more rigid and more formalized. Sometimes they are even modeled after data consistency algorithms. The resulting protocols are more adapted to computers than to users. Their abstract concepts need to be translated into notions that can be manipulated by users, usually through software layers and user interfaces. This leads to paradoxical situations where users have to coordinate themselves through multiple computer layers, and thus interact with each other by manipulating the concepts of the computer instead of their own human concepts.

A consequence of such designs is to induce extra manipulation costs for users to get access to the protocol management layer. For instance, shared drawing tools allow multiple users to manipulate the same drawing. In order to protect graphical objects from parallel modifications, they provide handles or even a lock button that can be added to an object to gain property rights on it. Similar mechanisms can be found in conferencing systems where floor control policies are used to designate a floor-holder who is allowed to manipulate the shared windows. In the same way, role attribution is another coordination protocol that requires explicit manipulation from users. All these manipulations can distract users from their main goal.

In addition to manipulation costs, designing the computer as an intermediate leads to coordinations that are much poorer than their natural equivalent. Our introductory example from ATC illustrates such losses. As short as they may be, conversations between controllers are negotiations, in which voice intonations or hesitations may be very meaningful. These extra messages convey social meaning, like stress, or availability. They allow controllers to adapt their strategy, for instance by making counter proposals if they think their colleague can manage, or by simplifying his task if they perceive a serious problem, thus improving security. Replacing them with

dialogue boxes eliminates that wealth of additional information. One can thus wonder if generalizing the use of computer protocols is always necessary, or if human natural protocols should be used instead when explicit coordination is not needed.

3. COORDINATION WITHOUT CSCW

When people do not use groupware tools, they use their own coordination policies based on their social skills. A quick review of these skills suggests new ways for supporting coordination.

3.1. Implicit Coordination Protocols

In the physical world, we also use coordination policies: we take turns when talking and avoid collisions when walking on a crowded pavement. At a more sophisticated level, we use coordination policies that we call social rules, like politeness or honesty. These rules may be very subtle. For instance, politeness often requires innocent lies. A very innocuous example consists in not answering your phone when you are busy talking with someone. At the opposite, efficiency or security sometimes requires impoliteness: when urgent attention is requested, we do not hesitate to break into conversations. Such examples show how hard it is to formalize coordination policies, and what kind of losses formalization can bring. Conversely, it is true that we sometimes refer our actions to explicit protocols that are comparable to computer protocols. But this is limited to specific situations such as testifying in court or signing a contract. In most situations, we use our "natural" social skills. We call them **implicit protocols**.

Some of those implicit protocols are supported by the communication space. Studies on turn taking in face to face meetings have shown that eye gaze is a clue used to distribute speaking time among participants. In the same way, voice intonation or body movements also contribute to the coordination. This suggests that *in real life, communication cannot be dissociated from coordination.*

3.2. Coordination Objects

When communication is not sufficient to ensure an implicit coordination of people's activities, the set of objects present in the environment can suggest some implicit protocols. Most objects have a social signification and support complex social rules. For instance, you know that you must not search the drawers of your office-mate, except if she or he is out of town and gives you a phone call asking to do so. Like drawers, many objects support social rules, but impose them only up to a point. We call them **coordination objects**.

Unlike their electronic counterparts, such as lock handles, coordination objects are not dedicated only to coordination. They belong to the production space. and share the same characteristics as the artifacts produced in common. They can be exchanged between people. They can also represent abstract notions. For example a key can be seen as an access right. In the same way, a hand set during a phone conversation can be seen as an "half-communication" which can be given to a third person. Objects and their manipulation are the real world equivalent of the production space in computer applications. This suggests that *in the real world production cannot be dissociated from coordination.*

4. DIRECT COLLABORATION

The previous intuitive analysis of coordination suggests that coordination needs not to be explicit to be efficient. It even appears that implicit coordination can be very effective and flexible when adequate support is offered by the communication and production components. This is what we call direct collaboration. *A direct collaboration system is a collaborative system in which coordination between users is supported by communication and production tools, and not by dedicated coordination tools.*

Direct collaboration can be compared with direct manipulation (Shneiderman, 1983). The latter was introduced when most applications provided command languages that users had to master in order to interact with their data. Command languages acted as an intermediate between users and tasks, like coordination protocols do between users. They introduced manipulation costs in the same way. Direct manipulation was introduced as an incentive to eliminate that intermediate. It was associated with requirements such as the continuous presentation of the objects of interest, with immediate feedback for actions.

Direct collaboration suggests that explicit coordination in groupware systems be replaced with adequate communication and production, which would then play the same role as presentation and actions in direct manipulation. In both cases, the computer has to be a medium rather than an intermediate. In direct manipulation, it is a medium between a user and virtual objects. In direct collaboration, it is a medium between users, with virtual objects as part of the medium as physical objects are part of our environment.

Even the limitations of direct collaboration can be compared to those of direct manipulation. In single-user interfaces it is sometimes necessary to constrain the user to a limited choice of actions, because a strict procedure has to be followed. In groupware systems, this corresponds to the situations when rigid protocols are necessary and cannot be ensured through implicit social rules.

5. DESIGN RULES

Direct collaboration appears as a fertile support for analyzing groupware. But can direct collaboration systems be really designed, and how? In order to use people's natural skills for implicit coordination, a groupware application should not filter out the clues on which they base that implicit coordination. We have defined three rules that can be applied to design systems featuring a more direct collaboration between people.

5.1. Integrating Communication Media

Communication channels whether natural (e.g. voice) or electronic (e.g. a connection between shared windows or a telepointer connection) are not always easy to access in the interface. When switching a task from foreground to background attention, it is often desirable that media are more integrated so that the groupware system can manage them and avoid extra manipulation costs. For instance such manipulations are required to send a fax during a phone conversation, when a single phone line is available. This implies at least five different steps: decide who will call back, hang up, dial the fax number, send the fax, then make a phone call again. It could be avoided with multiplexed voice and data.

The definition of a session in a groupware system involves the same kind of manipulations. The definition of a session is sometimes required to be able to connect together shared applications. Only then, the documents of interest can be imported into the session, through the selection of the session among the list of available sessions. Another solution would be to start the session directly from the document or to have an icon representing the session and to drop it onto a document to share it.

Natural communication channels, especially when their manipulation cost is negligible according to the task, provide effective coordination support. For example, informal observation of wargame players who have concluded an alliance, but play in different rooms, indicates that a voice link is better than a chat box to communicate information and coordinate actions. Giving a phone call implies a manipulation cost that is negligible for game players. However, in another situation, this would be perceived as a disruption in the task. The prototype described in section 7 shows how to integrate phone calls with other objects from the production space.

5.2. Integrating Activities

During collaboration, background and foreground activities are run at different paces, and from time to time they are synchronized or interchanged. These synchronizations are natural coordination points, as they integrate threads of activity into one main stream of collaboration. To avoid unintended disruptions of a task, groupware systems can manage pending tasks and let the choice of when to collaborate to the users.

In mediaspaces, permanent video links between different places help people to find appropriate times to get each other's attention, and to switch collaboration from a background activity to a foreground activity. When two persons can see they are both present, with the help of a mediaspace, they can decide to start a desktop videoconference, launching extra applications like a shared drawing tools if necessary. The mediaspace interface can be used to leave information accessible when a person is not available, like calendar browsing, e-mail, or notes to leave them a message (Tang & Rua, 1994). Likewise, a coordination object can be left on somebody's workspace to remind her or him of doing something with no interruption of the current activity. For example a shared workspace can be associated with each user and receive objects representing pending requests such as a "phone call" button.

5.3. Production Space as a Medium

The two rules described above have sometimes been addressed by previous literature. They mainly correspond to an improvemement of communication channels. But groupware also provides us with another communication channel: the production space, in which digital objects are being exchanged and manipulated. The production space can be used as a channel for conveying social hints in the same way as traditional communication channels. This is possible by introducing interaction styles that support **prosody** in the same way as voice intonation or gestures accompany oral communications, thus reinforcing coordination hints.

Whether working in a real or digital setting, users fill their environment with the objects they produce or manipulate during their activity: documents, pens, or other artifacts. These objects easily become part of the communication space when they are referred to in a discussion: "Give me this paper, please". They can also become part of the coordination space when one manipulates or displaces them. Seeing what happens and how it is done provides information about how to react: depending on how far and how quietly a document is pushed into your private space, you will stop what you are currently doing or not. As cartoon animation shows: objects and their manipulation can indeed carry a form of prosody, just like voice carries prosody.

In (Bentley *et al.*, 1992), the observation of air traffic controllers shows that physical manipulations of paper strips representing aircraft routes convey information to controllers working side by side.

Coordination objects, where, when and how they are manipulated and the standard social rules combine to suggest implicit protocols.

6. OBJECT EXCHANGE

To explore how objects from the production space can be integrated into the coordination space, we have developed several experimental prototypes. We designed them so that they convey implicit coordination cues for their users. We call them **Transfolders**, a contraction of two words: transmission and folder.

6.1. Transfolders

Any graphical object like an icon is a potential coordination object. By moving it in someone's space, it can be used for catching her or his attention. If that space is filled with other objects, representing communication channels for example, they can be used to start communications. For instance, URL objects could be dropped onto a URL viewer object to open a window showing that URL onto someone else's screen. Many transmission protocols, like email or remote folders, do not provide any information about the moment at which the document or the message transmitted is accessed. We defined transfolders so that they provide that feedback.

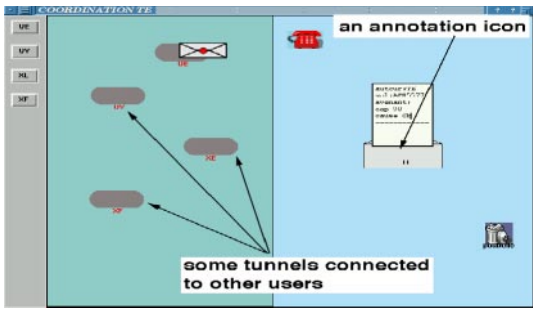

Figure 2: A shared transfolder for an ATC research prototype

A **tunnel** is a replicated place in an interface. Each replica is an end of the tunnel and can be located in different users' applications. When a user drags an object onto a tunnel end, that object becomes visible to any user looking at the tunnel end. When a user removes an object from a tunnel end, that object disappears from every visible end of the tunnel. To enforce that feedback, the object cannot be used as long as it is visible inside a tunnel. The left part of figure 2 shows four grey holes which are tunnels connecting a user with four other users so that they can exchange annotations, represented by an envelope icon.

A **shared transfolder** is a replicated workspace in an interface. Anyway, by convention it is associated with one user and represents a personal space.

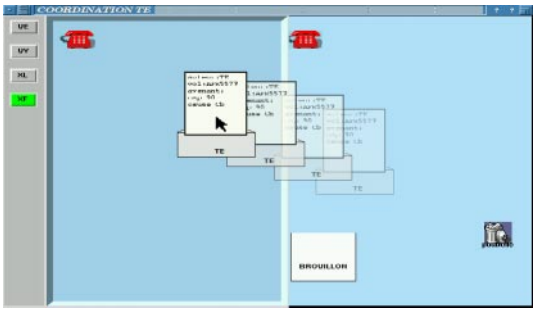

Figure 3: A transfolder connected to another one, side by side

Actions occurring inside a shared transfolder are replicated when at least two shared transfolders are connected together. To access a user's shared transfolder, one needs to connect his or her own transfolder with that user's transfolder, as shown on figure 3. Then they can exchange and manipulate their objects, and disconnect their shared transfolders when terminated (their transfolder then looks like figure 2, with the tunnels on the left side). Section 7.2 describes our application of transfolders to ATC. In this prototype, we have not implemented URL and URL viewer objects, but we use a phone object in quite the same way. Dropping an annotation on the phone icon gives a phone call to the owner of the icon.

6.2. Transfolders and Shared Workspaces

Like shared workspaces, shared transfolders are persistent stores for the objects they contain. They can be used either in synchronous (more than one person at a time) or asynchronous mode (different persons at different times). But they differ from shared workspaces on a few points:

- they are limited to a small portion of screen space as they are more oriented towards document exchange than towards in-place edition of shared documents. Thus workspace navigation is reduced to the connection to the transfolder of another user, and not to the synchronization of user's view (to maintain the What You See Is What I See property)
- access control is made simple as edition of objects occurs in the user's personal space and not inside a shared space. Shared objects are replaced by mobile objects exchanged between users.
- connections between shared transfolders are "lateral". In our prototype, each shared transfolder represents a different place. When connected together, these places are just aligned border to border so that an object can be dragged from one to another.

When accessing a user's shared transfolder is intrusive (for example if the transfolder contains personal annotations), a second public transfolder can be used instead. That public transfolder is visible to

everybody and acts as a repository for messages between all the group members. For example in a computer-supported meeting room, Courtyard uses a large display common to all users (Tani *et al.*, 1994). A “lateral” connection between the border of each user’s private display and the main display allow them to exchange objects.

Tunnels are limited to object exchange. They increase the visibility of the transmission in a way similar to the Pick-And-Drop metaphor (Rekimoto, 1997). With the Pick-And-Drop metaphor, the tunnel is replaced by a stylus (a physical device) which holds the data until it is dropped onto a screen sensitive to the stylus.

7. EXAMPLE APPLICATION

As we explained in the introduction of this article, our aim is to improve the collaboration tools used by air traffic controllers. In order to avoid the limitations of the dialogue box shown in figure 1, we propose tools that improve the temporal efficiency of collaborations without degrading their flexibility and their support for traditional social policies.

We took a two-step approach based on the design rules described in section 5. The first step was to propose a compromise between an improved efficiency and the preservation of the directness of collaboration. The first prototype described below proposes the integration of a voice link with an ATC workstation through our own computer-telephony integration layer. In the second step, we applied the integration of activities rule. This required the definition of new ATC objects that could serve as coordination objects. These objects have been integrated in the workspace of controllers through a transfolder metaphor. Following the third rule, the next step will consist in further developing the role of the computer as a medium by using the manipulation of these objects as a support for implicit coordination.

7.1. Integrating Media: DuoPhone

As explained above, in order to keep the role of phone calls as a support for collaboration, our first step was to add communication capabilities to a prototype ATC workstation. For that purpose we used Grigri, a gesture-based prototype of ATC workstation developed at CENA (Chatty & Lecoanet, 1996) and extended it with our DuoPhone communication software. We called the resulting prototype GriPhone[1]. Figure 4 shows the physical layout of GriPhone.

For our voice channel, we used the ISDN capability of the Sun SPARCstation 10, managed through our dedicated software layer DuoPhone. This allowed us to use a high quality voice link (digital telephone), with no constraints on the computer network. This is also important for backward compatibility with traditional communication channels: with DuoPhone, a controller can call any control room in the world. DuoPhone, developed at CENA in 1994, works as a server. It manages the ISDN link of the workstation, and allows client applications to give, answer, and transfer calls, connect the microphone and loudspeaker of the workstation to the phone link. It also allows them to subscribe to events such as incoming calls or disconnections. DuoPhone also monitors the telephone or any device connected to the same ISDN outlet, so that end users may freely choose between their favorite device: telephone handset or computer microphone and loudspeaker. Finally, DuoPhone allows applications to manipulate the facilities offered by ISDN: caller ID and messages. Therefore, when an application gives a call, it can decide to send a short text message with the call. If the recipient of the call is another computer running DuoPhone, client applications are informed of the origin of the call and the contents of the message, and thus can decide how to handle or display the call.

Figure 4: GriPhone: a telephone link controlled through the same interface as the ATC functions.

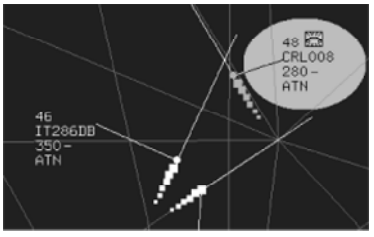

Figure 5: Someone is calling about aircraft CRL008.

Using DuoPhone allowed a seamless integration of voice communications into the main user interface of our ATC workstation. In the case of GriPhone, we just had to enrich the gesture library with a new gesture for giving phone calls. Upon such gestures, GriPhone just needs to decide whom to call, and send

1.*Griffonne* is a French verb meaning *scribble*.

the corresponding request. This integration of phone calls into the graphical interface becomes really significant when using the message associated to each call. When the gesture used to give someone a call is drawn on the representation of an aircraft, GriPhone interprets it as a call related to that precise aircraft. It thus passes the ID of the aircraft along with the call. Therefore, the other GriPhone workstation that receives the call knows which aircraft is concerned, and can display it (see figure 5). As the call is displayed next to the relevant aircraft, controllers can start the phone conversation without using the voice link for locating the aircraft that caused the call.

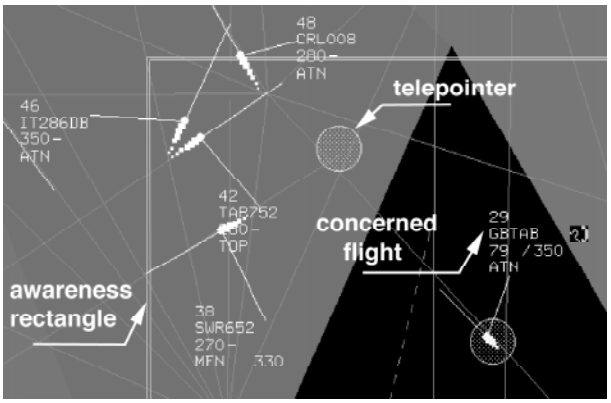

Figure 6: A communication between sectors.

Once voice communications are smoothly integrated into the system and calls are initiated from the display, further uses of the display as a communication channel become natural. We chose two services responding to a request from controllers to improve mutual awareness. The first service is a rectangle that shows what part of the display is seen by the other controller (see figure 6). This helps controllers to understand what their counterpart knows. The second service is a telepointer, adapted to the use of touchscreens. The telepointer is represented as a colored disk. Each controller has a different telepointer and can move it on both screens with a finger. With those services, GriPhone becomes a communication channel, through which users can talk and show things to each other. Shared sessions initiated from a phone call do not disrupt controllers from their main activity, which is centered around the radar display.

GriPhone has been informally presented to more than 20 air traffic controllers. As a large majority of them reported us their enthusiasm about the integration of communication media and the comfort it would bring them, we asked an evaluation of a similar prototype to another team. That formal evaluation, not described here, indicates an improvement of 17% in the time taken for a coordination using the phone when the incoming phone call is shown by an icon next to the relevant aircraft instead of not displaying any information about the relevant plane. More information about this evaluation can be found in (Karsenty & Pecaut, 1998).

7.2. Coordination Objects for ATC

As mentioned above, we believe that voice communications, because of their flexibility, are important to security and efficiency of communication. However, the number of phone calls is sometimes perceived as too large, and many air traffic controllers ask that routine phone calls be eliminated.

For that purpose, our second step was to define coordination objects. We chose to introduce objects that controllers can exchange through transfolders to integrate theirs activites. In addition to being perceived as a “physical” support for coordination, such objects also had to be relevant to the task of individual controllers. We analysed the ATC activity from that specific point of view, compared collaborations in different european ATC systems, and identified potential candidate objects at three levels of abstraction:

- aircraft are the most obvious objects referred to in dialogues. However, controllers mention aircraft or point at their representation, but do not exchange them.
- flight plans, or more generally contracts with pilots, are more abstract but more pertinent: controllers read them and are notified of changes in the contracts.
- the most pertinent objects are alterations of contracts, such as clearances. Even though these objects have no concrete existence as of today, they are well understood by controllers and essential to their activity.

Most manipulation tasks performed by air traffic controllers can be reinterpreted as manipulation of contract alterations, and collaboration tasks as their exchange:

- taking notes after giving an instruction to a pilot creates and archives an alteration
- preparing an instruction creates a draft alteration
- presenting a choice or a proposal to another controller consists in transmitting a draft alteration, that can be confirmed, modified and sent back, or built together.

This suggests different designs for creating and editing alterations, as well as for exchanging them. For instance, the dialogue box of figure 1 is a possible design for exchanging a flight level alteration. But applying direct collaboration metaphors we described, alterations can also be made first class visible objects. This is what we did, by giving a representation to contract alterations and using transfolders to exchange them (see figures 2 and 3).

7.3. Production Space as a Medium

With this design based on transfolders and icons, controllers will be able to choose their coordination styles, and even their degree of synchronization. Dropping an alteration in a remote corner of the working area will probably mean a low need for synchronization, while one dropped next to the phone icon in the transfolder will be interpreted as a request for a phone call. We hope to observe emerging working conventions, which would give us hints that we provided controllers with a new medium they can tailor to their needs.

(Bentley & Dourish, 1995) discuss the idea of supporting collaboration versus providing coordination mechanisms embedded in the heart of the system. They propose to provide a medium through different levels of customisation. Each level, from parametrisation of interface and of deeper system features to attachment of scripts to system events, allows users to invent new protocols. Our approach is different because, unlike customisation, which needs to be expressed in a form understandable by the system, prosody needs not be understood by the system. Prosody just needs to be carried by the system, its interpretation being left to users. Creating that sort of systems requires enriched feedback and a better control of users on actions and their transmission.

8. CONCLUSION

In this article we have outlined a potential problem of groupware systems that impose rigid coordination protocols on users and prevent them from applying their social skills for implicit coordination. As a consequence the computer is perceived as an intermediate rather than a truly collaborative medium. We introduce the notion of direct collaboration interfaces, focusing the design of interfaces around rules that could lead to a better integration of collaborative activities with other activities. As in the physical world, this integration can be improved by manipulation of adequate coordination objects. Once identified and implemented into the system, these objects can be used to integrate communication channels so that people can start collaborations directly from the relevant objects with a minimum number of manipulations. They can be used to integrate activities so that people can start their collaborations at the best time according to their activities. The manipulation and the circulation of coordination objects can support the exchange of implicit coordination clues, thus increasing the ability of the production space to be a medium. To demonstrate the feasibility of direct collaboration interfaces, we have presented some prototypes and metaphors we are applying to ATC workstations. Beyond ATC, we expect that many other domains will benefit from the same design rules.

ACKNOWLEDGMENTS

We would like to thank the persons who have contributed to Griphone: Patrick Lecoanet, Frédéric Lepied, Jean-Luc Dubocq.

REFERENCES

Bentley R., Hughes J., Randall D., Rodden T. , Sawyer P., Shapiro D. and Sommerville I. (1992). Ethnographically-informed systems design for air traffic control. *Proceedings of ACM CSCW'92*, pp123-129.

Bentley R. and Dourish P. (1995). Medium versus mechanism: supporting collaboration through customization. *Proceedings of European Conf. CSCW'95*, Kluwer Academic Publishers, Dordrecht, pp133-148.

Buxton W. (1992). Telepresence: integrating shared task and person spaces. *Proceedings of Graphics Interfaces'92*.

Calvary G., Coutaz J. and Nigay L. (1997). From single-user architectural design to PAC*: a generic software architecture model for CSCW. *Proceedings of the ACM CHI'97*, Addison-Wesley, pp242-249.

Chatty S. and P. Lecoanet P. (1996). A pen-based workstation for air traffic controllers. *Proceedings of the ACM CHI'96.* Addison-Wesley, pp87-94.

Ishii H. and Kobayashi M. (1992). Integration of inter-personal space and shared workspace: Clearboard design and experiments. *Proceedings of ACM CSCW'92*, pp33-42.

Karsenty L. and Pecaut I. (1998). Évaluation d'un collecticiel d'aide aux coordinations dans le contrôle aérien. Technical Report NR98-798, Centre d'Études de la Navigation Aérienne, 1998.

Rekimoto J. (1997). Pick-And-Drop: a direct manipulation technique for multiple computer environments. *Proceedings of ACM UIST'97*, pp31-39.

Shneiderman B. (1983). Direct Manipulation: a step beyond programming languages. *IEEE Computer*, pp57-69.

Tang J. and Rua M. (1994). Montage: providing teleproximity for distributed groups. *Proceedings of ACM CHI'94*, pp37-43.

Tani M., Horita M., Yamaashi K., Tanikoshi K. and Futakawa M. (1994). Courtyard: integrating shared overview on a large screen and per-user detail on individual screens. *Proceedings of ACM'94*, pp44-50.