Proceedings of the Third European Conference on Computer-Supported Cooperative Work
13-17 September, 1993, Milan, Italy
G. De Michelis, C. Simone and K. Schmidt (Editors)

Design for unanticipated use.....

Mike Robinson[1]
University of Århus, Denmark

Abstract: Support for work practice is better conceptualised as support for activity taking place in a multidimensional space than as prescription of temporal task sequences. The notion of "common artefact" is introduced to illustrate, unify, and summarise recent research that identifies significant dimensions of cooperative work. Common artefacts may be mundane, everyday objects like hotel keyracks or sophisticated computer tools Both are multidimensional, in that they provide orthogonal features. They are predictable; help people see at a glance what others are doing (peripheral awareness); support implicit communications through the material being worked on; provide a focus for discussion of difficulties and negotiation of compromises (double level language), and afford an overview of the work process that would not otherwise be available. It is argued that CSCW should support these dimensions of work, rather than trying to anticipate its specific sequentiality.

Introduction

Computer systems and applications that mediate work between people are increasingly discovered to be used in ways that were not anticipated by their designers. The paper highlights themes that emerge from the last decade of work in CSCW that help explain this, and which provide a framework of sensitizing concepts for the design and evaluation of future systems. Failure to anticipate sequences of action is explained by recent ethnography, e.g. Button (1993), that demonstrate order — and hence sequence — to be an ongoing product of people's work. Work is thus best supported by the provision of resources. The search for abstracted optimum sequences is understandable given the successes of "scientific management"

[1]The author would like to thank the COMIC ESPRIT Basic Research Project for supporting this work, and the COST 11 COTECH Working Group IV (CSCW Design) for discussion and comment on many of the ideas

(Taylor, 1947), and the sequential nature of most computing machinery and programming representations. The following three examples illustrate support for specific activities is best designed to avoid the anticipation of sequence.

Information Lens (Malone et al., 1987a; Malone et al., 1987b) was initially designed to provide an "intelligent" agent, an automatic secretary, to filter electronic mail. One well publicised function was to get rid of junk mail before it was read, or even seen. In a field study, Mackay (1988; 1990) noted that several people were unwilling to use this function. They felt a need to be aware of messages, and to view everything, at least at a cursory level. Of the many modes of use devised and shared by users, the one of special interest here is an innovation where users ran the rules (the way in which Information Lens sorted e-mail) on their messages *after* they had read or scanned them. Users needed to impose their own sequence: deciding between a secretary-filter to sort mail on the way in, or a secretary-archiver to put mail in appropriate folders after it was read.

Second, in a large field study of a number of "groupware" products, Bullen and Bennett (1990) noted many "fancy" functions of these packages were ignored. In particular, the sequentially structured conversational model and message categories in The Coordinator (Version 1) was largely by-passed. Many of those interviewed reported they ignored the choices, and just "hit enter" to send a message. They found messaging and selective archiving useful, but needed to create their own messaging sequences.

A third interesting example is given by Schmidt and Robinson (1993), citing Kaavé (1990). In a company with 50% of the world market in specialised optical equipment, production was controlled by a Manufacturing Resource Planning system (MRP). The system "knew" about all the products, sub-assemblies, and components down to the last nut and washer. It "knew" about the route each product took through the various production shops, and the number and breakdown of labour hours. From this, and other information, it calculated a production master schedule. As Schmidt observes, such a system is only feasible under conditions of limited product range in a stable market, so deviations in sales can be countered by inventories of products. Shortly after the MRP was introduced, the factory changed from large scale manufacture of a limited product range to order-driven production of a wide range of customised products. Production was driven by local plans based on current orders, not the master schedule. Yet the MRP system continued to be used. The workers found out that by using it 'backwards', it gave them an invaluable overview of the whole work process — letting them see, for instance, whether parts they would need shortly were scheduled.

These examples illustrate unanticipated use. In each case the designers had specific sequences of operations in mind. In each case the users appropriated the functions, while drastically reconstructing the planned sequences to match their actual work. This does not mean that CSCW systems design cannot be informed by analysis of practice: only that the practice is better conceptualised in a multi-dimensional space rather than as temporal task sequences. In the following sections, current understanding of the nature of work will be briefly summarised; "common artefact"

characterised as a multi-dimensional support system consistent with this understanding; and several relevant dimensions of work practice illustrated. It will be concluded that, in CSCW, the search for common artefacts is a better starting point for analysis and design than a search for work sequences.

The nature of work...

Unanticipated use of computer artefacts reflects the fact that work itself is underdetermined until realised in situ. Empirical and theoretical work over the past few decades has shown that there are severe limits to the "programmability" of work, and to the project of precise "Scientific Management" [2].As Suchman (1983) put it:

> "While for computer scientists "procedure" has a very definite technical sense, for practitioners of office work the term has some other more loosely formulated meaning and usefulness. The distinction is something like that of a predetermined and reliable sequence of step-like operations versus an unelaborated, partial inventory of available courses and desired outcomes."

Suchman (1987) later used the term "situated action" to underscore the view that every course of action is essentially *ad hoc,* and depends on material and social circumstances. The function of plans and other abstract representations

> ".... is not to serve as specifications for the local interactions, but rather to orient or position us in a way that will allow us, through local interactions, to exploit some contingencies of our environment, and to avoid others."

Gerson and Star (1986) made a similar point with the notion of "articulation work"

> *"Without an understanding of articulation, the gap between requirements and the actual work process in the office will remain inaccessible to analysis. ..* It will always be the case that in any local situation actors "fiddle" or shift requirements in order to get their work done in the face of local contingencies. We argue here that *such articulation* is not extraneous to requirements analysis, but *central to it.*"

None of this means that organisational procedures, structures, roles, workplans, objectives, and so on, are without value. Local flexibility, articulation work, and the shifting of requirements all happen in order to get the work done, within a framework of plans and objectives. It is simply that no procedure, no anticipated sequence of events, will ever match the rich, concrete detail of an actual situation. Procedures are more like advice than algorithms. For most purposes, such forms of organisational memory help make good choices and avoid pitfalls. But, as March (1976) points out, the ability to forget and overlook is essential.

Other factors in the nature of work impinge on attempts to anticipate use of artefacts. Schmidt and Bannon (1992) point out that working relationships are embedded within larger ensembles. They may be transient, and where they are not, patterns of interaction change with the requirements and constraints of the situation. Membership is not stable, and may even be non-determinable. Putting this in other words, there is no way of telling in advance exactly who will be needed where or

---

[2]See, inter alia, March and Olsen, 1976, Mintzberg, 1979, Salaman, 1979; Winograd and Flores, 1986, Anderson et al., 1987, Bjerknes et al., 1987, Suchman, 1987; Galegher et al., 1990; Schmidt, 1990; Robinson and Bannon, 1991

when... The consequence for CSCW design is that neither sequences of action, nor specific actors are pre-determinable. What can be known in advance is summarised in the phrase *"fluid transitions"*..... There will be fluid transitions between individual and cooperative work (Hughes et al., 1991), between formal and informal interactions, between different tasks (Reder and Schwab, 1990), and between different media and tools (Bignoli et al., 1991; Ishii and Arita, 1991). Specific work is often an attempt to link required activities, to prevent the needs of one task from disrupting another, or to replace or repair a missing resource. Workflow diagrams and other types of formal representation may be a resource, or a frame for understanding work. They cannot predict or prescribe such specifics.

"Common artefacts" neither anticipate sequences of actions, nor attempt to enforce procedures. Like a Common in a traditional village, they allow many patterns of use without needing to anticipate specific actions or people. Common artefact has its origin in an attempt to understand the role of language in CSCW applications (Robinson, 1991a). It was noted that "successful" applications seemed to allow two modalities of communication: natural, fairly unrestricted conversation and communication via a system where actions were constrained by formation and transformation rules. A good example is an outliner, used in collective authoring (Ellis et al., 1991). Here it is clear that participants need to create and change a structure using rules, and also need to discuss their work. Both modalities of communication are necessary, and together are termed "double level language". "Common artefact" is an elaboration of the dimensions of communication that take place through, and are supported by a system.

Common artefacts

> *"The altitude of a technology might not be measured only in terms of the sophistication of the inner workings of the hardware device, but also in terms of the extent to which the device renders an important problem easy to solve"*
>
> (Hutchins, 1990)

Mundane everyday life provides many examples of common artefacts. Many of them work so well that they are taken for granted, inconspicuous, and appear trivial. Most of them are not considered to have any great "technological altitude" or sophistication. Yet they contain important lessons on the dimensions of computer applications to support people working together.

The keyrack behind the reception desk in a hotel is an example. Guests can leave and collect their keys; can see which other guests are in or out, and leave messages in the pigeonholes. Hotel staff use it to communicate with their colleagues, and place bills, faxes, etc. to be given out to guests. The presence of keys, or the contents of pigeonholes, conveys information, and may be the subject of questions or discussion. Some operations are considered legitimate, while others are not: usually only the receptionist can place keys or messages; keys have to be hung over appro-

priate numbers; etc. The keyrack is a model of the hotel, mapping the rooms. A glance at the it in the late evening gives an overview of the hotel occupancy.

Yet this keyrack is not foolproof, nor is it "active". There is nothing to prevent keys being hung in the wrong places, or lost. It can be used in many idiosyncratic ways — probably violating the recognised procedures. Conversely, there are certain things about a keyrack that are fixed, like the positions of the hooks. It is simply not possible to hang a key between two hooks, as there is nothing to hang it on. So potential uses are a result of physical properties, local conventions and rules, and situated activities.

It is argued that the mundane, old fashioned keyrack summarises the dimensionality necessary for Computer Supported Cooperative Work. It has a further advantage of being comprehensible to designers and users alike. The following sections will explore generic dimensions of common artefacts that appeared in the example.

Characteristics of common artefacts

Predictability

Common artefacts need to be predictable (and hence dependable) to the people using them. At this level, they are simply a tool for getting the job done — the keyrack stores keys. Examples in CSCW change train times, negotiate wages, move passengers and baggage between planes, steer large ships, coordinate air traffic, and schedule the prescription of medicines in hospitals.

In order to function as a tool at all, an artefact needs structure and operational predictability (the two notions are not independent). Hooks that did not stick out would be troublesome. Spreadsheets that did not enforce a grid structure would be useless. Even the universal tool[3] (Bannon, 1989) depends on structure for its multiplicity of uses. No-one wants an unpredictable artefact, whether it is a machine or a computer program. Predictability is probably the best understood aspect of system design, in terms of functions to be provided, consistency and compatibility between them, and appropriate human interfaces.

It should not be forgotten that there are many designs for tools to do any particular job. The particular form of any given artefact is a consequence of processes of competition, persuasion, selection, evolution, and diffusion. There is no Platonic Blueprint for a keyrack. Its general form has evolved because it makes an important problem "easy to solve". The important problem is not just storing keys — there are ways of storing small bits of metal that do not take up several square metres of space. Understanding the nature and dimensions of the "important problem" that everyday common artefacts resolve can reveal areas that CSCW applications need to address if they are to achieve equal success.

[3] An old fridge handle that was used as bottle opener, hammer, tea stirrer, etc.

Predictability (function, dependability, appropriate interface) is a crucial part of CSCW application design. But there is a danger of over-concentration on this, to the exclusion and detriment of other dimensions of cooperative working.

Peripheral Awareness

Implementing basic functionality alone can lead to disruption of the work. A good example is the French Nuclear Power Station (Kasbi and Montmollin, 1991) where old-fashioned dials were to be replaced with better tools — more sophisticated computerised displays. These were easier to read for each operator individually. It was forgotten that the power station operatives, collectively, needed to be aware of what the others were paying attention to, and what they were doing. It was easy to see which part of the system was of concern when someone went over to a dial and stared at, or tapped it. It was easy to read the context of any action that might subsequently be taken. This would have been lost with the "better" individual displays, which were fortunately not implemented.

Hotel keyracks and the power station setup allow experienced people to see what others are doing "*at a glance*" (Goodwin and Goodwin, 1993). This is possible because both artefacts and operations on them can be seen easily. "Peripheral awareness", and its importance for collaboration is illustrated by several studies ranging from libraries (Anderson and Crocca, 1992) to Air Traffic Control Rooms (Bentley et al., 1992). A good example is provided by Heath and Luff (1991):

A London Underground Control Room

The Bakerloo Line control room can house several staff. Two main actors are the Line Controller and the Divisional Information Assistant (DIA). The latter, amongst other things, provides information to passengers via a PA system, and to Station Managers by touch-screen phone. Both are able to see the state of Bakerloo line traffic on a real-time display which runs the length of the room. There is a radio system for contract with drivers, a PA control panel, and closed circuit television for viewing platforms. In addition, a paper timetable specifies train numbers, times, and routes; crew allocations, shifts, and travel; vehicle storage and maintenance; etc. The timetable is actively used by all control room staff to identify difficulties with, and manage the service.

> "Controller and DIA cover their paper timetables with collophane sheets which allows them to mark changes and add details with a felt pen and later to remove the various arrows, figures, and notes. ..... the various changes undertaken by the Controller are rarely explicitly told to DIA or to others (who) pick up the various changes (and) sketch in the reformations and adjustments on their own timetable."
>
> (Heath and Luff, 1991)

The layout and the evolved constellation of common artefacts support peripheral awareness. This facilitates some near-telepathic coordination between Controller and DIA. For instance, to keep trains running at regular intervals, the Controller was observed to ask a driver on the Charing Cross Southbound line to delay for a

couple of minutes. The DIA, glancing at the real-time display, aware of the timetable, and overhearing the controller's first attempt to contact the driver, is announcing the delay to passengers before the controller has finished his request to the driver.

Such situations illustrate that, in addition to predictability and functionality, a dimension of peripheral awareness will often need to be taken into account in the design of common artefacts. If this is not supported, damage to coordination may easily outstrip any gains.

Implicit Communication

Another dimension of common artefacts is implicit communication, though as will be seen later, this is inextricably bound up with double level language. Hotel staff use keyracks to communicate with their colleagues, for instance by leaving bills. In other words, they communicate without (necessarily) exchanging words. They communicate *implicitly* through their actions on the common artefact. This feature of work was first noticed by Pål Sørgaard, who used the term "shared material".

> "A simple example is the way two people carry a table. A part of the co-ordination may take place as explicit communication, for example in a discussion of how to get a table through a door. When the table is carried, however, the two people can follow each others actions because the actions get mediated through the shared material. This co-ordination is not necessarily explicit."
>
> (Sørgaard, 1988)

Carrying a table through a door is an excellent but limited way to illustrate *implicit communication*. When people carry out a joint activity, conversation is only the tip of an iceberg. Much communication is mediated by the material they work on, whether it is physical like the table, or symbolic, like accounts or portfolios. Schmidt (1993) noted potential disadvantages of this as a sole means of coordination:

(1) The bandwidth is usually very restricted.

(2) The turn-around time of the interaction is determined by the frequency of state changes in the artefact.

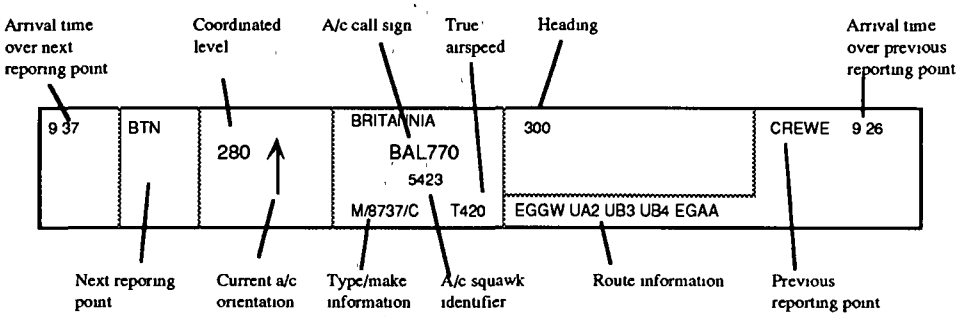
(3) The act is embodied in the state change in the field of work.

On the other hand, the restricted nature of implicit communication through an artefact has advantages — if other means, like talking, are not prevented. It augments the number and type of channels available, allowing parallelism and delayed responses not available in simple speech. The changes marked by felt pen on the cellophane timetable cover in the London Underground Control Room are one example. Implicit communication, reading states of work objects as signs, becomes more powerful, and less open to misreading, when accompanied by local convention. This is well illustrated by the following example.

Flight Strips in Air Traffic Control: An example of convention in implicit communication

A long term study on cooperative work in air traffic control has provided interesting insights into the delicate balance between the characteristics of artifacts and cooperative work practices (Harper et al., 1989; Harper et al., 1991; Bentley et al., 1992). Civilian flights over England and Wales are controlled from a center near London. Airspace is divided into 16 sectors, each of which is controlled from a "suite", equipped with radar screens, TV monitors, telephones and other communication facilities, maps, computers etc. Typically, a suite is staffed by two air traffic controllers, two assistants, and one sector chief. Controllers have three main artifacts to aid them. Radar displays show a trail of 'blips' representing a particular flight, with a data block alongside showing the flight number and flight level. Telephone and radiotelephone links enable controllers to talk to pilots, other controllers, and neighboring airspaces. Flight progress "strips" contain information on each flight. These latter items are the focus of this investigation into the nature of common artefacts.

The strips are made of card, approximately 200 by 25 mm, and are divided into fields. Information in the fields comes from a database holding the flight plan filed by the pilot prior to departure, sometimes modified by inputs keyed in by controllers or assistants. It includes the aircraft's callsign, flight level, heading, planned flight path, navigation points on its route, estimated time of arrival, departure and destination airports, and the aircraft's type. Strips are arranged in racks immediately above the radar screens. The collection of strips enables a controller to gauge how many aircraft are due in the sector, where they are bound and when, and the strip can be used to record any instructions given to the aircraft.



A flight progress strip.

When a controller gives an instruction to a pilot, for example to ascend to flight level 220, he or she marks this on the strip. In this case, the mark is an upwards arrow and the number 220. When the pilot acknowledges, the controller crosses through the old flight level on the strip. When the new flight level is attained, the controller marks a check beside it. Changes in heading, estimated time of arrival, route, call sign etc. are dealt with in similar ways.

> "This means that *this* information, and any subsequent revisions, is 'ready at hand' as the 'current state of affairs'. As a controller aptly remarked, the strips 'are like your memory, everything is there'. Important to this functionality are the conventional notations on a known-in-common format and a known sequential organisation of the strips themselves. They are note pads, but not personal ones."
>
> (Harper, 1889)

Each team member uses a differently coloured pen so the source of the annotations is immediately apparent. The changes in state of the common artefact are *conventional and formalised* (upward arrows, numbers, check marks, crosses through numbers). Any old mark or annotation will not do. The signs have to be immediately readable and comprehensible. This is a clear example of how convention supports implicit communication.

In similar vein, anyone who notices a problem with a strip or pair of strips — perhaps two flights due at the same navigation point at a similar time and at the same height — can "cock out" the strips, i.e., move them noticeably out of alignment in the racks. This makes it immediately obvious that, when it becomes time to deal with those flights, a problem will need to be considered, and to the practiced eye it will be obvious from a glance at the strips what the problem is. The vocabulary and syntax of annotation and ordering the strips are a language through which the members of the team communicate with each other and create a 'common statement' about the state of the flight and of the sector (Hughes et al., 1992).

Changing flight levels and "cocking out" strips are both cases where a common artefact becomes articulate because its state can be "read" according to convention. The artefacts support this since they are predictable; can be taken in at a glance; and additionally, by providing a set of symbols for which there are formation and transformation rules, they support conventionalised implicit communication. This is the further dimension of cooperative work that needs to be taken into consideration.

Double Level Language

Implicit communication can only happen when the participating actors are able to *maintain* an evolving set of rules, understandings, and expectations about the meanings of actions, signs, and changes of the common artefact — in other words, when the *participants can also communicate directly using the fullness of their natural language to interpret the concrete situation in front of them.* This is a further dimension of cooperative working that needs to be facilitated, and certainly not prevented, by any common artefact.

The achievement of mutual learning, coordination, negotiated conventions, accident recovery, and misunderstanding repair has been examined by Gerson and Star (1986) and by Suchman (1987) with the important concepts of articulation work and situated action. Neither concept implies that coordination can be accomplished by speech *alone*. The concrete work situation and its common artefacts are critical in grounding conversation and spoken exchanges. They provide essential context

for explicit communication.[4] "Double Level Language" is a phrase intended to catch the idea that implicit, often indirect communication (through artefacts) and explicit communication (speech, ad hoc notes) are not alternatives, but complementary and mutually supportive.

In general it can be said that any non-trivial collective activity requires effective communication at both levels. 'Computer support' is valuable insofar as it facilitates the separation and interaction between them. Applications that support one level at the expense of the other tend to fail. The dimension of implicit, formal or conventionally readable "states" is essential as it provides a common reference point for participants. A sort of 'external world' that can be pointed at, and whose behaviour is rule-governed and predictable. But this 'world' is meaningless without interpretation, without the talking that maintains its meaning. Conversely, dialogue is almost vacuous unless it is grounded (Robinson, 1991a: 43).

The Wage Bargainer is an example of the symbiotic relation between these two types of communication. It provides an 'external world' that can be pointed at, whose behaviour is rule-governed and predictable, and that facilitates and grounds dialogue and negotiation.

The Wage Bargainer[5]

The problem was two hundred people in a dozen or so workgroups, with a wage structure that had "evolved" over 19 years. During this time, the organisation had increased in size six fold, and many new specialisms and functions had developed. The wage structure had become so complicated that it was almost incomprehensible. Anomalies and discontent abounded. The organisation was a co-operative, so there was no external owner, or separate management to bargain with. No single group could *impose* a new structure. In practice, and in some way, the workers had to bargain with each other.

An Open University team, in consultation with co-op members, created a distributed spreadsheet. Each group could change any of the factors that directly affected it (wage scale and range, increments, overtime rate, number of workers, etc.). Although any group could make any change, what the co-op *could* pay was determined by the relation of the aggregated "claims" to total (budgeted) income. Finally it was agreed that the reality of the budget income was such that a 3% across-the-board increase was the minimum everyone would accept, and the most that could be paid. Anomalies remained, but the problem they were causing was resolved. The final agreement could not have been deduced from the original claims. It was necessary to have the iterative, dialectical movement between the formalised 'world' of the spreadsheet and the negotiations and discussions.

---

[4] Ethnomethodology and Conversation Analysis have demonstrated the *indexical* character of spoken exchanges. "This" and "that" are obvious examples of words that do not have meaning outside specific contexts. It now appears that the majority of human interchanges need an awareness of the particular situation in which they are happening in order to be comprehensible.

[5] developed in 1987 by the Open University as part of the WISDOM Project to explore collective uses of new technology (Robinson, 1988; Robinson, 1991b)

In CSCW, such reciprocity can be ignored. Some applications try to support conventional (formal) operations on the common artefact, while others concentrate on providing additional channels for dialogue. The success of either will depend on whether or not the people who use the system can *maintain both modalities* of communication.[6] Cooperative work needs to be both *discursive* (setting rules and conventions, achieving agreements) and *indexical* (grounded in the objects of work). Communication happens at both levels. The combination of implicit and explicit communication is "double level language". This is supported by mundane artefacts like keyracks, timetables, and flight strips. Computer applications may ignore this lesson. They may make it difficult to access the object of work, and disrupt indexicality and implicit communication. Alternatively, they may simply assume formal communications (signs on an artefact of some sort) are enough, and attempt to disallow discussion, resolution of ambiguity, further elaboration of context or detail. In both cases, the systems are generally disliked, not used, and, in the end, fail commercially (Ehrlich, 1987b; Ehrlich, 1987a; Grudin, 1989; Markus, 1990). Alternatively, as with the undoubtedly successful e-mail (Bannon, 1985), the application may support just one level of language, provided that the participants have an established and trusted method of communicating on the other level.

Overview

Common artefacts afford an overview on the work-world *which would not otherwise be available.* As an extra dimension, overview has greater spatio-temporal scope than peripheral awareness, and need not have the conversational qualities of a double level language. Artefacts that afford overviews have a long history. The usefulness of maps, photographs from the air, watchtowers, exploded diagrams, keyracks showing room occupancy, etc. does not need to be argued. Computers graphics are a powerful way of gaining overview on social trends and phenomena that would otherwise be "mere numbers". The appearance and importance of "overview" in computer-based artefacts that were designed to do something else is an exciting discovery of CSCW research, and well illustrated in the following study.[7]

Overview of the Ward in a Norwegian Hospital

Bjerknes and Bratteteig (1988) tell a story of the design and implementation of a system to support nurses in their daily work — "The Florence Project". The researchers implemented a "work sheet system" based on scraps of paper used in the short-stay cardiological ward. The nurses had previously used these scraps to make notes of the most important patient information at the beginning, update during, and

[6] The controversy over whether electronic mail is a CSCW application or not is related to this distinction. E-mail allows conversation-like exchanges, but does not support access (peripheral, implicit, or otherwise) to the object or context of the conversation.

[7] See also Schneider and Wagner (1992) for a similar overview effect in a study of information systems and collaborative work in four French hospitals.