Reflexive CSCW: Supporting long-term *personal* work

Harold Thimbleby[1]	Stuart Anderson	Ian Witten
Computing Science	Computer Science	Computer Science
Stirling University	Edinburgh University	Calgary University
Stirling	Edinburgh	Calgary
FK9 4LA	EH9 3JZ	T2N 1N4
Scotland	Scotland	Canada

June 21, 2000

Abstract

CSCW (Computer Supported Cooperative Work) is an active research area with many promising applications and benefits. We argue that the plight of the individual user can also be viewed as a CSCW problem, for the individual frequently acts as multiple persona: performing many independent tasks, perhaps in several places. We propose *reflexive* CSCW to address such issues. Solutions in the reflexive case will of course be of benefit to users even if they are working in a conventional multi-user CSCW context; proposed solutions in CSCW can be re-presented for individual users.

[1]Address for correspondence.

1 Introduction

As computers become more widely available and as their memory expands into gigabytes per user, users can be expected to lose track of their multiple activities mediated by the system. Despite computers having so much memory, they cannot readily be used to aid human memory. Whereas humans set up situations and objects in the world to remind themselves of future commitments—writing is just a special case; more often, people arrange papers on their desk, tell other people to remind them, put books in the doorway so that they will be seen at the right time, and so on ([Norman, 1988])—computer memory is bland and unable to support such cues.

CSCW (Computer Supported Cooperative Work) aims to facilitate cooperative work between numbers of people distributed in both place and time. The insight underlying this paper is that even a *single* person has work distributed in both place and time, and can therefore greatly benefit from extended support *à la* CSCW. Personal work is distributed in *place*, both internally within a hierarchical filestore, and externally because the user may be working perhaps at home and at work, with various backups and data in transit between sites. Work is distributed in *time*, for activities may be interleaved and run over periods of years. We therefore define **reflexive CSCW** to be the stance of CSCW recruited to personal work, whether as theories, systems or techniques.

There are many cases of conventional CSCW systems being used in a reflexive style (for example, users sending email to themselves to remind them of their future commitments): but this style is an accidental consequence of CSCW features being creatively redirected by users. Reflexive CSCW intentionally aims to design appropriate features for reflexive use. Obviously, reflexive CSCW can appropriate conventional CSCW technology; less obviously, new perspectives and applications are opened up by the reflexive stance—and it is those which this paper starts to explore.

Aims of this paper

This paper discusses and highlights problems arising when users attempt to maintain many interleaved activities over arbitrarily long periods. The aim of this paper, then, is to argue that there is a need for reflexive CSCW and, furthermore, that reflexive CSCW is both feasible and widely applicable.

This paper is intended to open up a new research area, to show how CSCW ideas can be profitably directed to bear on pressing personal computing and information management issues. The purpose is to get people thinking about personal work from the fruitful perspective of CSCW. The paper does point to new, and apparently powerful, techniques and interface ideas, but at this stage it is premature to do anything other than sow the ideas, which we do from both practical and theoretical perspectives.

2 Background

Hard discs of even modest capacity readily allow the personal long-term information stored to exceed the user's abilities to manage it and exploit the investment it represents; worse, many users today have access to several computers, and they may accidentally or deliberately have multiple copies of various versions of their work distributed between the systems—effectively unmanageable, certainly error-prone, without computer supported assistance.

A user's activities are represented by transactions with programs, mail, databases, printers, filestore structures and so on, over periods of years. Operating systems already time-stamp file activity, but this information is rarely analysed by the system. Such information, although implicit in filestores, is not normally directly available to the user—and certainly cannot be used as a means of access or navigation. At best, time information is *locally* analysed, at the level of recency ordering files within directories (i.e., within predetermined spatial clusters).

As an experiment, we implemented a simple system showing *global* recency, and this was indeed very useful after periods away from the computer, when the user's work context had been lost to him as a result of cognitive interference with unrelated events. The program simply listed all the user's activities in most recent order; it was surprising that such a simple program was so effective! Another system we implemented runs at night, performing numerous (user-specifiable) checks on files that have been created or modified during the day; it then produces a comprehensive report for the user on the last day's activities.

Simple cluster analysis and temporal access methods would enable users to recover information, by association with either related computer-based activities or external activities easily located in time.

A deliberately *simple* scenario illustrates the potential. An access tool might display a chosen month from a calandar, with colour used to indicate space–time correlation. The user clicks on a 'bright' activity, and is taken to the corresponding context—the user might have been away for a week, has forgotten details of his outstanding activities, and now wants to resume what he was doing a week ago last Tuesday. The presentation of the data would be customised to cater for differing work cycles (e.g., shift, week, term, quarter), and could be filtered or transformed in various ways to contrast certain activity classes (e.g., projects this week, projects related by keyword, critical projects). Note that such schemes are in principle independent of the activity unit; it is however convenient to illustrate reflexive CSCW using familiar objects such as files and file-based activities, rather than in terms of more abstract activities, as may be represented by correlations between collections of files and operations.

Future directions

For the future, we would aim to develop systems based on a sound theoretical model of temporal action and commitment that will support individual (and loosely-coupled group) work on many concurrent computer-based projects; we briefly review some of the relevant theoretical work below (in Section 6). Such a system should be able to provide assistance in recording and meeting commitments, monitoring the progress of projects, identifying and grouping related activity, and investigating alternative scenarios, all of a reflexive nature. It should provide facilities for the user to construct externalised salient cues, linking or attached to activities and contexts. It would also be possible for the system to learn to identify patterns of activities that the user is likely to label as cues at a later time. We have already noted such cues (and their support) are necessary but lacking in conventional interactive systems.

3 Supporting reflexive CSCW

Reflexive CSCW is intended to be a new way of looking at supporting personal work. It is therefore premature to expect a description of techniques for supporting reflexive CSCW at this stage, but this section gives specific illustrations of how reflexive CSCW can influence actual interface design.

Current systems provide inflexible access methods, based either on a name space or a spatial metaphor. Both approaches are inflexible, the structures imposed become obsolete, and effective navigation and location of old objects relies on the user maintaining an accurate model of the structure. The problem is greatly exacerbated the smaller the user's task unit, particularly in hypertext. In a filestore, the unit of activity is a file; in hypertext the unit of activity is a node (or card), and the user spends less time at each unit of activity. "Getting lost" is much easier in hypertext than in conventional filestores, though when it happens in filestores the consequences may be more severe as more information is involved.

While user interface problems with within–session navigation in large databases are widely recognised, the management of personal activities over long time scales involves rather different problems. Aspects of these problems have been identified in specialised and restricted ways in IPSEs (e.g., project management, source code control), databases (navigation), hypertext (ideation, "getting lost"), and of course in CSCW. Reflexive CSCW unifies these separate and specialised views of the problem. In particular, we stress the non-invasive potential for a more general approach—consider that the project management facilities of an IPSE are explicit, normative, and specialised to the programming domain. The limitations of invasive, normative interfaces have been noted elsewhere ([Carasik, 1988]).

Winograd ([Winograd, 1987/8]) takes a language/action perspective for his *Coordinator* ([Action Technologies, 1987]), a program which allows one to record the commitments associated with items of electronic mail. Insofar as the *Coordinator* and similar systems represent CSCW, reflexive CSCW is as if all work was treated by the *Coordinator* as mail to oneself. But we are concerned not with interpersonal communication (as in mail) as such but with non-trivial reflexive communication; such communication arises almost uniquely in large personal computerised workspaces over periods of time.

4 Other applications

1. Window based systems are increasingly *persistent*: windows are may persist from one session to the next. At present display clutter and the difficulty of managing many windows limits users from fully exploiting persistence. In fact, the problems are very similar to the problems

identified for managing large filestores, except, of course they become apparent with far less information.

2. Fully interactive applications, represented by window managers, must allow for the stored information to change other than by the user's personal agency. Mail may arrive, a program may produce long waited-for results. For such issues, conventional CSCW is appropriate. But for a persistent window system: simply substitute *window* for the reflexive CSCW *file*; we believe the support reflexive CSCW provides is largely independent of the activity unit, whether window or file.

3. When a single user works on more than one computer (perhaps a desktop machine and a portable, or perhaps a desktop machine at work and one at home) his problems are multiplied disproportionately: in addition to managing the extra disc memory, he has to remember where projects are located, which versions are current, and which slave—and what to reconcile if he is forced to work on an obsolete version (i.e., because the current version is on the other, inaccessible, machine). No current systems allow the user to communicate with his other personae (his other machine at the other site), for instance to remind himself to obtain a copy of a file. Yet such facilities are trivial CSCW: and, indeed, directly suggested by the reflexive CSCW idea.

4. Finally we note that there are an increasing number of proprietary 'work organisation' software products appearing on the market, addressed at personal work organisation, and offering integrated diary and planning facilities. Suffice it to say that this is market recognition of the issues raised in this paper; yet these commercial products are clearly ignorant of the CSCW approaches and resources that could have been recruited.

5 Secondary advantages of reflexive CSCW

Reflexive CSCW provides the following less obvious bonuses, whatever the unit of user activity:

- Enhances security: reflexive CSCW systems can locate and track work activity, and can do so whether these are caused by legitimate use (as in window management) or by unauthorised access, such as viruses.

- Reflexion and its theoretical models are in principle domain-independent, so are quite general, applicable through: electronic mail organisation; program version control; document control; program browsers; to aspects of large-scale industrial projects. This would allow investment in developing the ideas and, signficantly, the cost to users of learning the technology to be amortized over many applications.

- A successful reflexive CSCW system is likely to have a radical and positive impact on work habits: it would encourage more exploratory modes of work, since there is no risk of 'lost work'; it would ameliorate the effect of staff turnover, since new staff would be able to rapidly familiarise themselves with previous activities and outstanding commitments. Reflexive CSCW would certainly help clarify personal commitments and so make working, particularly in groups, more efficient.

6 Theoretical perspectives

Theoretical frameworks in the field of human computer interaction are somewhat unpopular: it has quite rightly been pointed out that much more 'up front' advances are required before abstract theory will make a major impact on user interfaces. However, we include this section about the theoretical perspectives of reflexive CSCW because of an important issue: generality.

Reflexive CSCW is in principle a general concept. It is not relevant what the unit of the user's activities is, whether it is a file, hypertext node, or a window. It therefore follows that beneath the concrete representation of the reflexive CSCW facilities lies a common structure. If this structure can be analysed abstractly, that is independently of the particular choice of activity, then theoretical results will apply to all sorts of activity. Furthermore, understanding the 'simple' issues in reflexive CSCW—where only one user is involved—will help underpin analogous issues in conventional CSCW—where many users are involved, possibly simultaneously.

All user interfaces make *ad hoc* use of notions of time, place, resource constraints, and so on, to reduce the burden of information provision on the user. A central problem is the construction of systems in which these features are treated—but not palliated!—in a uniform, systematic framework.

Theoretical frameworks—principally speech act theory and temporal logic, at different levels—already exist within which the user's activities can be modelled. For example, speech act theory identifies just a few forms of felicitous commitment which are held to be of universal application. This suggests that the novel forms of reflexive systems theory of the sort we are proposing could be both *simple* and *universal*.

The whole range of "non-standard" logics offers many insights which can be exploited to support long-term work. Temporal logic ([Rescher, 1971]) offers a framework for reasoning about events and their temporal relation. The family of relevant logics ([Dunn, 1986, Girard, 1987]) seem promising as a way of reasoning about resources. Recently temporal logics have been discussed in connection with artificial intelligence ([Allen, 1983, Allen, 1984, McDermott, 1982])—a discussion that would have benefited greatly from a rigorous ontological characterization of approaches to describing intervals and events such as that of Van Bentham ([Van Bentham, 1983]). Anderson's work ([Anderson, 1988]) combining modal logics with user interface considerations is highly relevant. Jefferson's virtual time ([Jefferson, 1985]) provides an invaluable general mechanism for allowing users to backtrack during their work, undoing tentative commitments (for instance: mail sent but not yet opened; programs modified but not yet compiled) where possible if they so desire. The idea of global time advancing independently of local time and lagging behind it provides a natural and general way to solidify commitments and to explain to the user why they can no longer be undone.

Our current theoretical work is using the mu calculus ([Anderson & Thimbleby, 1990]).

7 Summary

People readily lose track of the implicit constraints acting on their set of projects. Computer disk store may hold many years' worth of work on several projects, yet conventional filestores provide few (typically just one) structuring mechanisms. This problem has not previously been identified as separable: it is clear that theories, techniques and systems can to a very large extent be machine and application independent, applicable to a wide range of systems.

There have been many attempts to provide an informational backbone for CSCW. Teleconferencing systems, mail systems, 'participant' systems ([Chang, 1987, Stefik, 1987]), information-sharing systems ([Malone, 1987]), have all become popular topics for research, not to mention the facilities provided by routine project management software (many proprietary systems) and programmers' workbenches. We believe our vision for reflexive CSCW is more strongly rooted in the diversity of practical reality than these efforts. It is more ambitious because it aims to provide an infrastructure for working with multiple projects and could do so unobtrusively. Our idea of reflexive CSCW is more rooted in practical reality because it can be built around the user's present environment, using his *regular* file space (or whatever) as a base; rather than being a 'project organiser' that expects him to re-orient his way of working to suit the new tool.

Acknowledgements

The authors wish to thank the anonymous referees for their helpful comments which have helped us improve the paper.

References

[Action Technologies, 1987] Action Technologies Inc., "The Coordinator workgroup productivity software", Emeryville, California, 1987.

[Allen, 1983] J. F. Allen, "Maintaining knowledge about temporal instances", Communications of the ACM, **26**, pp832–843, 1983.

[Allen, 1984] J. F. Allen, "Towards a general theory of action and time", Artificial Intelligence, **23**, pp123–154, 1984.

[Anderson, 1988] S. O. Anderson, "Interface semantics: A linguistic approach", IEE Colloquium on Formal Methods in Human Computer Interaction II, Digest 1988/82, 1988.

[Anderson & Thimbleby, 1990] S. O. Anderson & H. W. Thimbleby, "Towards a Framework for Computer Supported Collaborative Work", Department of Computing Science, Stirling University, 1990.

[Carasik, 1988] R. P. Carasik & C. E. Grantham, "A case study of CSCW in a dispersed organisation", in CHI'88, pp61–65, 1988.

[Chang, 1987] E. Chang, "Participant systems for cooperative work", In *Distributed artificial intelligence*, edited by M. N. Huhns. Pitman, 1987.

[Dunn, 1986] J. M. Dunn, "Relevance Logic and Entailment", in: D. Gabbay, F. Guenther (eds.), *Handbook of Philosophical Logic*, Vol III, pp117–224, D. Reidel, Dordrecht, 1986.

[Girard, 1987] J-Y. Girard, "Linear Logic", Theoretical Computer Science, **50**, pp1–101, 1987.

[Jefferson, 1985] D. R. Jefferson, "Virtual time", ACM Trans Programming Languages and Systems, **7**, pp404–425, July 1985.

[Malone, 1987] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst & M. D. Cohen, "Intelligent information-sharing systems", Communications of the ACM, **30**, pp390–407; May 1987.

[McDermott, 1982] J. McDermott, "A temporal logic for reasoning about processes and plans", Cognitive Science, **6**, pp101–155, 1982.

[Norman, 1988] D. A. Norman, *The Psychology of Everyday Things*, Basic Books, 1988.

[Rescher, 1971] N. Rescher & A. Urquhart, *Temporal logic*. Springer-Verlag, 1971.

[Stefik, 1987] M. Stefik, D. G. Bobrow, G. Foster, S. Lanning & D. Tatar, "WYSIWIS revised: early experiences with multiuser interfaces", ACM Trans on Office Information Systems, **5**, pp147–167, 1987.

[Van Bentham, 1983] J. F. A. K. Van Bentham, *The logic of time*. D. Reidel, Dordrecht, Holland, 1983.

[Winograd, 1987/8] T. A. Winograd, "A language/action perspective on the design of cooperative work", Human Computer Interaction, **3**, pp3–30, 1987/8.