Applying Reflection to CSCW Design

Paul Dourish
Rank Xerox EuroPARC
Cambridge, UK

dourish@europarc.xerox.com

1 Introduction

"And since you know you cannot see yourself
So well as by reflection..."
— Shakespeare, Julius Caesar, Act 1 Scene 2.

This paper outlines work in progress concerning the application of computational reflection to the design of systems for Computer-Supported Cooperative Work (CSCW). A meta-level architecture for the design of a CSCW toolkit provides a novel approach to adequate provision of flexibility in the design of these systems. In particular, it points towards the ability to support dynamic flexibility in group behaviour, which is critical to the usability of such systems.

2 CSCW Systems

Interest and research in CSCW systems has grown considerably over the past five to ten years. A basic premise of this field is that much work activity is inherently collaborative in nature. Traditional computer systems, however, tend to isolate their users from each other, and, in the cases where resources are shared, create the illusion that a user has sole access to them, hiding the activities of others. CSCW systems make the sharing and cooperation of individuals explicit and provide direct support for processes which involve multiple individuals.

The range of potential CSCWapplications is vast, and their designs vary widely. Systems may be synchronous (that is, all users are simultaneously co-present in some work space) or asynchronous (that is, the work object is handed back and forth between the individuals); they may be focussed on informal or formal interactions; they may provide varying degrees of information about the activities of individual participants or the group as a whole; they may support a variety of roles for participants, reflecting their activities or working relationships; they may provide different edit control mechanisms, and provide shared working over text, graphics or active data such as a dynamic database or spreadsheet.

So, CSCW systems support a wide variety of activities and work styles. In their design, flexibility is a key requirement. We can point to three major classes of flexibility in such systems:

1. *Static* flexibility is a form of flexibility in a system which does not change during a work session, or changes only slowly from session to session. Examples of static flexibility include interface-specific tailorings made by individuals, and tailoring by groups; accommodation of different working styles, which might be imposed by individuals, groups or organisations; and the need to adapt to other parts of the surrounding environment (*e.g.* compatibility with existing single-user tools). Some of these issues are described in [Greenberg, 1991].
2. *Dynamic* flexibility involves responding to changing aspects of the group and the group's behaviour. Group behaviour will change a great deal in the course of a collaboration, and such changes are frequently swift and tacit. The activity of individual participants also changes as group work progresses, as roles and responsibilities are renegotiated, and as members switch between individual and group work. Group membership will also change, with individuals joining and leaving the group. Inflexible support for group dynamics, and its impact on the collaborative process, are problems cited in studies of computer-supported collaboration [*e.g.* Austin *et al*, 1990].

Position paper for ECOOP92 Workshop on Reflection.
Last modified: April 14, 1992

3. *Implementational* flexibility covers the adaptability of a system to a variety of underlying infrastructural and architectural changes. For instance, the network architecture might dictate a choice of centralised, distributed or replicated architecture for the shared application; and this might need to change with the dynamics of the group. While these changes may often be minimal on the surface (*i.e.* at the interface level), they are major reorganisations of the internal state of the system.

3 CSCW, Flexibility and Toolkit Design

The range of potential CSCW applications, and the flexibility required of them, are major problems for the designer of a CSCW toolkit and may, perhaps, account for the small numbers of such toolkits. The problem, of course, is the traditional one associated with the design of toolkits (or, for that matter, programming languages): implementation decisions made by the toolkit designer will limit the range of applicability of the toolkit. Existing toolkits, such as MMConf [Crowley *et al*, 1990] and Rendezvous [Patterson *et al*, 1990], are typically quite limited in their scope.

One example of the sort of decision which has typically been the responsibility of the toolkit designer concerns the issue of data replication in the architecture of the collaborative system. Some systems take a *centralised* approach, in which there is a single copy of some shared work object (*e.g.* a textual document or graphic), which resides on some server; all edit clients must communicate with the central server to make changes, so that the server serialises requests and ensures consistency for each client. Other systems take a *replicated* approach, in which each client has a private copy of the work object and can make local edits[1]. From an architectural point of view, the primary advantage of a replicated architecture is that network load is reduced since only input events need to be distributed to other participants; the primary disadvantage is that more complicated protocols must be used to ensure that inconsistencies cannot arise.

However, this architectural decision has important implications for the use of the collaborative system. Centralised architectures work well in the presence of fast, local-area networks; but for large numbers of clients, or collaborations across slower long-haul networks they become inappropriate; the server becomes a bottleneck, and the slow exchange with the server means that local changes are echoed slowly. Thus, the architectural decision concerning data replication has an important impact on the way in which the systems can be used and the settings in which they might be appropriate.

In a toolkit, this sort of decision is typically made by the toolkit designer; and thus, the architectural impact is not merely on a single application, which could be designed for one particular setting, but for any system built using this toolkit.

4 Reflection and CSCW Toolkit Design

In order to design toolkits which are applicable in a wider range of settings and which can be used to build more flexible applications, we are looking at the application of computational reflection to specify a meta-level architecture for a CSCW toolkit. Thus, applications built with this toolkit would contain within them a model of their own implementation which would be amenable to examination and manipulation. By defining the basic objects (or metaobjects) of the system, *i.e.* the terms in which its self-description is cast, and the relationships between them, we effectively produce a specification language for CSCW architectures and implementations which can be used by toolkit designers to build more flexible implementations.

Before proceeding to discuss some aspects of a reflective model of CSCW design and an example of the way in which it extends the functionality of current CSCW design models, another flexibility issue appropriate for CSCW will be discussed.

5 The Problems of Dynamic Flexibility

The issue of *dynamic flexibility* was raised in section 2 as one of the ways in which CSCW systems need to be adaptable. Dynamic flexibility is a particular issue in group work.

Observations of group behaviour, either in face-to-face meeting situations or in more long-term collaborative work such as co-authoring, reveal the extent to which

1. Other approaches, such as partial replication or migratory objects, are also possible; we consider merely the fully centralised and fully replicated implementations as an example of the impact of architecture on applicability in a variety of settings.

the activity of the group and its make-up change in the course of a collaboration. In a meeting, we encounter various modes of behaviour which the group might move through over a short period of time, including:

- *formal management*, characterised by control through a designated chair, the use of an agenda to structure activity, and so on;
- *informal group activity*, such as free-for-all brainstorming, where the group is focussed on some particular task but group members all contribute in a relatively unstructured way;
- *individual work*, where individual group members (or sub-groups) may work on ideas, documents or suggestions on their own before bringing them before the group.

Changes from one mode to another, furthermore, are frequent and often tacit. Co-workers develop mechanisms for managing not only the process of group work, but also the way in which it will change in the course of a collaboration. Traditionally, in as much as one can discuss a "tradition" in the context of such a young field, CSCW systems are very much focussed not only on one particular task, but one particular aspect of that task. Thus, the systems embody a notion of group process which typically deals with only one of these modes of working. Clearly, this means that either the system cannot support the entire group process, or that group process must change to accommodate itself to the support offered by some tool. Neither of these are particularly acceptable solutions. We would wish instead that CSCW systems can dynamically adapt themselves to the current state of group work and provide adequate levels of support for each.

Similarly, it is not only the group process which changes over time, but also the nature and composition of the group itself. Two particular manifestations of this will serve as examples. The first relates to the membership of the group. Group membership is not always clear in collaborative work (*e.g.* in co-authoring, do colleagues who comment on a draft count as members of the group for the purposes of group support?) and indeed, different group members may have different views of the current membership of the group. Further, even the "core" group of accepted members may change over time—in a survey of co-authors, Beck reports that 22% of groups changed membership even after the writing phase had begun [Beck, 1992]. The second manifestation concerns the use of "roles". Roles define the relationships between group members, and of individual members to the work at hand; essentially, they can be regarded as responsibilities for aspects of the work. Again, the assignment of roles is typically a flexible and fluid process; responsibilities are not prescribed, but negotiated. In CSCW systems which provide explicit support for roles, the dynamic nature of role assignment is often ignored. Thus, we would wish CSCW systems to accommodate themselves to the changing membership and nature of the group itself.

Clearly, dynamic flexibility is a requirement for a much wider range of systems that merely those supporting collaborative activities. However, in CSCW, support for the dynamic aspects of group and process organisation are critical to the adoption and usefulness of tools. We are, then, investigating the ways in which a reflective CSCW model, providing support for dynamic reconfiguration, can provide support for these aspects of group activity.

6 Dimensions of Flexibility

The work described in this position paper is at an early stage; I have been discussing primarily a framework within which our research is being conducted.

A primary focus for our work is the identification of dimensions of flexibility with CSCW systems. The basis of this work is a mapping of the design space for collaborative tools through the use of the QOC design rationale notation. This work, being carried out jointly with Victoria Bellotti, is focussed on the iterative reformulation of design questions and options, organised into informal groups which reveal the structure of the design space for a range of tools.

Currently, our work is principally driven from user and task support requirements. A major area which this has led to is the provision for information, provided through the workspace itself, for information about the character and content of the activities of other group members [Dourish and Bellotti, 1992]. As well as this support, and the issues discussed earlier concerning data replication, other important dimensions include:

- *degree of linkage between individuals*: systems can provide support for individuals sharing a workspace at the level of basic information, representations, view of information or presentations. The level at which the information is shared, as well as the degree to which these linkages can be changed for the group or for individuals, is an important architectural issue with strong implications for the use of the tool by groups;

- *edit control mechanisms*: collaborative systems can provide a variety of mechanisms for ensuring integrity and consistency of information, as well as means of, explicitly or implicitly, attaching group process information to data such that access follows a negotiation process.
- *presentation of synchronous and asynchronous information*: while CSCW systems are typically either synchronous or asynchronous in nature, we are looking at the way in which a flexible system can act semi-synchronously, providing both modes of access as appropriate. The issue of representations of synchronous and asynchronous activity information, and the degree to which they might be similar, different or open to adaptation, is another important part of our design space.

It is from such dimensions that we intend to form a reflective model. The dimensions will describe the space of systems which can be generated from this model, and the reflective model will provide programmers and users with a flexible approach to systems design, embodied in a toolkit which should be more widely applicable than is currently possible using traditional techniques.

Our approach, then, has three components. First, the development of the QOC design space; second, the identification and description of dimensions of flexibility for CSCW design; and third, the reification of these into a metaobject protocol. The metaobject protocol is used first as a descriptive tool, which helps determine its completeness, and then as the basis of the design of particular systems and toolkits.

7 Summary

Much of common work activity is inherently collaborative in nature, and the support of collaborative activities by computer systems is an active area of research. However, the design of such systems is a non-trivial task, due to the nature of group activity and the variety of approaches to the problems of group work. Designing a generic toolkit for collaborative systems, then, is an even more difficult task.

Our current research involves the application of the theories and techniques of computational reflection to the design of a toolkit for collaborative systems. By developing a metaobject protocol which describes the behaviour of collaborative systems we hope to provide programmers with a more flexible way to create systems. Problems of dynamic flexibility are a particular requirement for collaborative systems, and these are an important focus of our work.

More generally, when we consider systems which have to adapt to changing circumstances which are typically controlled socially, this raises questions of the appropriate definition of the meta-level. Such systems include with uncertain elements which can not be directly integrated into a computational model. It is our hope that, in the course of this work, we will be able to throw some light on those issues.

Acknowledgments

I would particularly like to thank Victoria Bellotti, Gregor Kiczales and Wendy Mackay for fruitful discussions of these ideas. The CSCW architectures project is being conducted jointly with Victoria Bellotti.

References

[Austin *et al*, 1990] Laurel Austin, Jeffrey Liker and Poppy McLeod, "*Determinants and Patterns of Control over Technology in a Computerized Meeting Room*", Proc. ACM Conf. on Computer-Supported Cooperative Work CSCW90, Los Angeles, Ca., October 1990.

[Beck, 1992] Eevi E. Beck, "*A Survey of Experiences of Co-authoring*", in Sharples (ed.), "Computer Supported Collaborative Writing", Springer-Verlag, forthcoming.

[Crowley *et al*, 1990] Terry Crowley, Harry Forsdick, Paul Milazzo, Ray Tomlinson and Ellie Baker, "*Research in Real-time Multimedia Communications Applications*", Report No. 7325, Bolt, Berenek and Newman, Inc., Cambridge, Mass., May 1990.

[Dourish and Bellotti, 1992] Paul Dourish and Victoria Bellotti, "*Awareness and Coordination in Shared Workspaces*", EuroPARC Technical Report, March 1992.

[Greenberg, 1991] Saul Greenberg, "*Personalisable Groupware: Accommodating Individual Roles and Group Differences*", Proc. European Conf. on Computer-Supported Collaborative Work ECSCW91, Amsterdam, September 1991.

[Maclean et al, 1991] Allan MacLean, Richard Young, Victoria Bellotti and Thomas Moran, "Questions, Options and Criteria: Elements of Design Space Analysis", Human-Computer Interaction, 6 (3&4), pp 201–250, 1991.

[Patterson *et al*, 1990] John Patterson, Ralph Hill, Stephen Rohall and Scott Meeks, "*Rendezvous: An Architecture for Synchronous Multi-User Applications*", Proc. ACM Conf. on Computer-Supported Cooperative Work CSCW90, Los Angeles, Ca., October 1990.