

Groupware Technology and Software Reuse

Position paper for the European Reuse Workshop (ERW'98).

Klaas Sikkel

Faculty of Computer Science, University of Twente
PO Box 217, 7500 AE Enschede, the Netherlands
sikkel@cs.utwente.nl

Computer-Supported Cooperative Work (CSCW) is a research field concerned with the design and implementation of systems to support cooperative work. Such systems are usually called Groupware. Although Software Reusability (SR) is not commonly mentioned as an issue in the CSCW community, there are some obvious overlaps in design issues and methodologies.

In this paper I will argue that reusability issues are of particular importance to groupware technology and relate our experiences in this matter.

Changing requirements

An issue of particular concern in the implementation¹ of groupware is that it is impossible to predict in detail how a system will be used, hence the requirements are at least partially unknown when the system is constructed.

Changing requirements is a general problem in software engineering, but in CSCW the problem is more grave than in other domains. In order to adequately support cooperation between people, one needs some understanding of *how* cooperation takes place. This is a very subtle and not fully formalizable process, as failures of early office automation systems and numerous CSCW studies testify [Suc83, Suc87]. Testing multi-user systems is problematic and can not well be done in a laboratory setting. In the CSCW community it is common knowledge that systems can only be evaluated *in situ*, at the workplace.

Case studies of successful groupware implementations show that systems evolve, as organisations learn to work with them [Orl96, BE96]. If the system is used in a given setting, it affects the way people work and enhances their "technological frame" [Orl92], i.e., their mental model of what the system can be used for, so they think of new ways to use the system. The technical system and the social system evolve in an interlocked process. (Note, however, that presupposes the system to be successful. Many CSCW systems have been installed but were never really used [Gru88]).

Consequently, the system has to be reworked over and over again; "maintenance" is not a proper word to describe this, "continuous implementation" is nearer to the mark. "Design for change" is one of the maxims of groupware systems design. But how to do so?

¹ "Implementation" is meant here in the broader sense of "getting a system to work within an organisation", not in the narrower sense of "creating executable software".

Design for change

In order to allow systems to change, several design and software engineering techniques have been proposed, which will sound most familiar to the Software Reusability community (but the term "Reusability" is never mentioned explicitly).

- Enhancing the facilities for end-user tailorability of systems [MC+90, HK91].
- Toolkits [RG96] and platforms [TRB93] that support easy construction of systems with (or without) a range of features. The most successful commercial example of a general platform is Lotus Notes.
- General tidiness of system design and documentation, so that the system can be adapted to changed patterns of use without undue effort [GSW].
- The use of existing, open standards, to allow easy interfacing with other systems and software [BH+95].
- Component-based groupware, finally, is called for more recently [tHo98, SC98].

In addition – and particular for the application domain CSCW – one may alleviate the problem by focusing on the type of support that is provided.

Tools that offer objects of some particular structure, relevant for a particular type of work, are more robust to changes in work practice than systems that support carrying out specific tasks [Rob93]. More radically, groupware systems provide a general *medium*, without incorporating any knowledge of what is it that people use it for, are more generally applicable than systems that incorporate, in one way or other, knowledge about the work being done [BD95]. An example of such a system is email; even though email doesn't satisfy most definitions of groupware, it is often quoted as "the most successful groupware application," the reasons for which are obvious from the above discussion.

Experiences with the BSCW Shared Workspace system

From 1994 to 1997 I was involved in a project that created a simple Web-based groupware system called "Basic Support for Cooperative Work" (BSCW)² [BH+95, BA+97]. We have particularly been concerned with the use of open standards and interfaces. Our prime concern was that the system should work in heterogeneous environments, but the pay-off in terms of software reuse was tremendous: *most parts of the system did not have to be built at all!*

BSCW has a client-server architecture; a BSCW server is an auxiliary component to an arbitrary WWW server (using the CGI standard for interfacing Web servers with application software). More importantly, there is *no* BSCW client; it can be used with an ordinary browser. Trivial as this may seem now, in 1994 this was not quite so obvious.

Most groupware is pretty complicated and you can invest a lot of person years in a sophisticated distributed runtime system before you get anything working on the screen. The small size of the initial BSCW implementation allowed us to reconstruct the system "from scratch" (or, more precisely, "from COTS") *twice* during the first 18 months. As

² <http://bscw.gmd.de>

things go, when exploring new terrain, the first system (and the zeroeth prototype preceding it) had not quite the right architecture and design.

So far the good experiences. In another case, integration with existing software proved to be mixed blessing. At some point it was decided to integrate the BSCW system with the Merlin process support environment [JP+94], so as to allow both structured and unstructured forms of cooperation within a single, distributed environment. A first prototype of the integrated system, "Process Support for Cooperative Work", linked the Merlin process engine to the BSCW user interface [SNS98]. This proved to be rather more complicated than initially envisaged, and plans for further integration of the systems were abandoned. In retrospect, this was a classical case of *architectural mismatch* [GAO94].

In order to do things right from a SE point of view, we used a CORBA-based interface package (ILU 2.0 [JS97]), forcing compliance with specified object interface definitions – or so we thought. It forces *syntactic* compliance. The semantics was harder than expected, due to the fact that both parts of the system "live in different worlds". Concepts which are self-evident in one environment are hard to grasp or make no sense at all in the other. One can very well live with the fact that one part has to fake activities required by the other part (e.g., having started a nonexistent process), but the semantics of these kinds of "senseless" operations are prone to subtle differences in interpretation – and the more so when the geographical distribution in the team coincides with the distribution of knowledge about the involved systems and their environments. Hence the protocol for exchanging the right messages at the right moment became more and more complex along the way and never became fully robust. To our experience, it is the speaking and thinking in different conceptual frameworks, more than the need to adapt and re-implement parts of the functionality, which makes it so difficult to integrate heterogeneous systems.

What we learned – the hard way – is the essential difference between creating a *new* system based on (a coherent set of) existing resources and integrating *existing* systems which lack a common conceptual foundation.

Plans for the future

At the University of Twente we are about to start a research project on evolutionary implementation of groupware. With this project, the issue is to be addressed from two different angles, both the managerial organizational aspects and the software engineering aspects of making systems evolve as the organization in which it is implemented evolves.

While our prime interest in the workshop is to learn from other participants more experienced in reuse, we believe that adding our own experiences and views from a CSCW point of view could contribute to enriching the workshop.

References

- [BA+97] R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor and G. Woetzel. Basic support for cooperative work on the World Wide Web. *International Journal on Human-Computer Studies* **46** (1997) 827-846.

- [BD95] R. Bentley and P. Dourish. Medium versus Mechanism: Supporting Collaboration through Customisation. *Proc. Fourth European Conference on Computer Supported Cooperative Work (ECSCW'95)*, Kluwer Academic Press, Dordrecht, Netherlands, 1995, 133–148
- [BE96] T.K. Bikson and J.D. Eveland. Groupware Implementation: Reinvention in the Sociotechnical Frame. *Proc. Conf. on Computer Supported Cooperative Work (CSCW'96)*, ACM, NY, 1996, 428–435.
- [BH+95] R. Bentley, T. Horstmann, K. Sikkel and J. Trevor. Supporting Collaborative Information Sharing with the World Wide Web: The BSCW Shared Workspace System. *Proc. Fourth International World Wide Web Conference*, O'Reilley & Associates, Sebastopol, CA, 63–74.
- [GAO94] D. Garlan, R. Allen, and J. Ockerbloom. Architectural Mismatch: Why Reuse is So Hard. *IEEE Software*, November 1995, 17–26.
- [Gru88] J. Grudin. Why CSCW applications fail: Problems in the design and evaluation of organization interfaces. *Proc. Conf. on Computer Supported Cooperative Work (CSCW'88)*, ACM, NY, 1988, 85–93.
- [GSW] P. Grefen, K. Sikkel, R. Wieringa. Two Case Studies of Subsystem Design for Extensible General-Purpose Software. Submitted for publication.
- [HK91] A. Henderson and M. Kyng. There's No Place Like Home: Continuing Design in Use. In: J. Greenbaum, M. Kyng (Eds.) *Design at Work: Cooperative Design of Computer Systems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991, p. 219-240.
- [JP+94] G. Junkermann, B. Peuschel, W. Schäfer and S. Wolf. MERLIN: Supporting Cooperation in Software Development through a Knowledge-Based Environment. In A. Finkelstein, J. Kramer and B. Nuseibeh (Eds.) *Software Process Modelling and Technology*. Research Studies Press Ltd., Taunton, Somerset, UK, 1994.
- [JS97] B. Janssen, M. Spreitzer. ILU 2.0 Reference Manual. Xerox PARC, Palo Alto, CA, 1997.
- [MC+90] A. MacLean, K. Carter, L. Löfvstrand and T. Moran. User-Tailorable Systems: Pressing the Issues with Buttons. *Proc. Conf. on Computer-Supported Cooperative Work (CSCW'90)*, ACM, NY, 1990, 175–182.
- [Orl92] W.J. Orlikowski. Learning from Notes: Organizational Issues in Groupware Implementation. *Proc. Conf. on Computer-Supported Cooperative Work (CSCW'92)*, ACM, NY, 1992, 362–369.
- [Orl96] W.J. Orlikowski. Improvising Organizational Transformation Over Time: A Situated Change Perspective. *Information Systems Research* **7** (1996) 63–92.
- [RG96] M. Roseman and S. Greenberg. Building Real-Time Groupware with GroupKit, A Groupware Toolkit. *Transactions on Computer-Human Interaction* **3** (1996) 66-106.
- [Rob93] M. Robinson. Design for unanticipated use *Proc. Third European Conference on Computer Supported Cooperative Work (ECSCW'93)*, Kluwer Academic Press, Dordrecht, Netherlands, 1993, 187–202.
- [SC98] O. Stiemerling and A.B. Cremers. Tailorable Component Architectures for CSCW Systems. *EuroMicro Workshop on Parallel Distributed Processing*, Madrid, 1998, IEEE Computer Society, Los Alamitos, CA, 302-309.
- [SNS98] K. Sikkel, O. Neumann and S. Sachweh. Process Support for Cooperative Work on the World Wide Web. *EuroMicro Workshop on Parallel Distributed Processing*, Madrid, 1998, IEEE Computer Society, Los Alamitos, CA, 325–331.
- [Suc83] L.A. Suchman. Office Procedures as Practical Action: Models of Work and System design. *Transactions on Office Information Systems* **1** (1983) 320–328.
- [Suc87] L.A. Suchman. *Plans and Situated Actions*. Cambridge University Press, Cambridge, UK, 1987.
- [tHo98] H. ter Hofte. *Working Apart Together: Foundations for Component Groupware*. Telematica Instituut Fundamental Research Series, No. 001, Telematica Instituut, Enschede, the Netherlands.
- [TRB93] J. Trevor, T. Rodden and G. Blair. COLA: a Lightweight Platform for CSCW. *Proc Third European Conference on Computer Supported Cooperative Work (CSCW'93)*, Kluwer Academic Publishers, Dordrecht, the Netherlands, 15–30.