Interacting with Computers vol 7 no 2 (1995) 195–210

Four principles of groupware design

Andy Cockburn and Steve Jones*

Groupware design is at a stage where identification, clarification and validation of best practice is critical if its potential is to be realised. The paper examines and records the major causes of groupware failure, and provides four groupware design principles that encapsulate the problems and guide design teams around them. The principles provide an extendable framework that is a synthesis of design lessons recorded in CSCW literature.

Keywords: human–computer interaction, groupware, design, princples, user-acceptance

The failure of early groupware systems is well recorded (Grudin, 1988; Grudin, 1990; Grudin, 1994). The design approach adopted in these pioneering development projects was often characterised by computer scientists intending to increase the efficiency of organisations radically through deterministic models of co-operative activity. Such design strategies have been shown to be inadequate: they fail to account for the social factors in group work. As a consequence, research into computer-supported co-operative work (CSCW) has broadened beyond computer science to include the social sciences that study the subtle factors encompassing collaborative work.

CSCW research focuses on a range of goals: from providing an understanding of the social factors involved in support for group work, to the development of 'point systems' that demonstrate the potential of new and innovative technologies. Evaluation and explanation of systems is frequently a casualty of the 'next system trap' with developers eager to eliminate current failings in their next implementation. Mistakes and misguided developmental decisions are frequently unreported, resulting in a lack of guidance for the next generation of system developers. Consequently the same, or similar, design errors are replicated and rediscovered.

The latest, and most promising, groupware development methodology is participative design (Muller and Kuhn, 1993). Interdisciplinary development

Department of Computer Science, University of Canterbury, Christchurch, New Zealand. Tel: +64 3 364 2774. Fax: +64 3 364 2999. E-mail: andy@cosc.canterbury.ac.nz. *Department of Computer Science, University of Abertay Dundee, Bell Street, Dundee, Scotland, UK. Tel: +44 1382 308619. E-mail: Steve.Jones@tay.ac.uk.

teams work with the end-users to co-determine the support they receive. Yet participatory design is no panacea. The misguided intuitions that caused the failure of early systems may be collectively held by the participatory design team. All involved need to share a common understanding of the issues relevant to groupware design. The principles described in this paper guide all those involved in design around the pitfalls that have been encountered, some repeatedly, by groupware.

Why principles?

In 1983, Donald Norman argued for "more fundamental approaches to the study of human–computer technology." He alerted human–computer interaction (HCI) researchers to the "tar pits and sirens of technology," referring to the temptations of system development and the self-serving enticements of new technology. He argued the need for fundamental principles to "broaden our views, sharpen our methods, and avoid temptation."

CSCW research is now in a similar situation to that of HCI in 1983. It can be viewed as research territory into which exploration has only just begun, rife with uncharted tar pits and sirens. The lessons of CSCW development are akin to folklore with design issues permeating local research communities, but often going no further.

The principles presented in this paper provide system designers[1] with a guiding 'chart', noting the relevant pitfalls, problems, and barriers to the development of successful co-operative work support tools. Personal intuitions, and a "feel for the right way to do things" remain a major part of design (and the key factor distinguishing good designers from bad ones), but intuitions, regardless of their foundations, are fallible. These principles alert designers to groupware's problems, and to the misguided intuitions encountered (some repeatedly) in collaboration support.

A pragmatic approach

The design guidance provided in this paper is derived from a number of sources. Much of it is gleaned from the fragmented lessons of groupware development that are recorded in CSCW literature. The social implications of groupware are often cited as the fundamental barrier to its success (Sproull and Kiesler, 1991). We do not argue with this, but rather, we note that a society's rejection of groupware is driven by an accumulation of individual rejections. If principles for groupware design make systems more acceptable to individuals, without hindering their value in group support, they are likely to be more acceptable to the user society.

Causes of groupware failure

Forming and maintaining collaborative relationships is difficult. Even in ideal

[1]In this paper we use the term "designers" to mean all those involved in (participatory) design. The shared understanding that is the aim of the principles is equally important to all participants, regardless of their educational background.

work environments continual trade-offs, give and take, between colleagues is required. The inclusion of computer support in this complex balance is often counter-productive. Rather than enhancing group efficiency and cohesion, computers hinder it. Typifying the extreme level of discontent with groupware, users called the *Coordinator* "worse than a lobotomised file clerk" (Carasik and Grantham, 1988), and reactions like the following about the Colab meeting support system are not uncommon:

> "... they found it so frustrating that they put their heads in their hands, raised their voices, and ultimately threatened to walk out." (Tatar *et al.*, 1991, p. 190)

What can be done to improve the poor performance of groupware? The issues of interest to groupware developers are:

- What can be done to avoid the failure of previous systems?
- Why have previous applications failed?

As a starting point, Grudin (1988) identifies three major causes of failure in CSCW applications[2]: the disparity between the person who does the work and the person who gets the benefit; the breakdown of intuitive decision-making in design; the underestimated difficulty of evaluating CSCW applications. These three points can be generalised into three levels of failure: system-use, system-design, and system-evaluation.

The following sections examine groupware failure in system-use. These observations provide a foundation for the principles that assist designers in avoiding failure in system-design. System evaluation is largely beyond the scope of this paper, but some comments on the role of principles in groupware evaluation are made in the discussion.

Effort inherent in collaboration

The costs or undesired aspects of collaboration are the overheads of *effort* beyond that required to execute personal work tasks.

Naturally, there are a number of social factors that can inhibit and discourage collaborative work, regardless of its supporting mechanisms. Although many of these factors can be considered to increase 'effort' (for instance, personality clashes make collaboration burdensome) such social complications will not be discussed in this paper.[3]

When collaborators are physically remote, communication mechanisms must be used to mediate the interaction. All non face-to-face interaction mechanisms are limited by their bandwidth which reduces the richness of interaction.[4] This necessitates greater *effort* in completely and accurately transferring information (Hollan and Stornetta, 1992).

---

[2]Grudin later extended these three points to include conflicts with social norms and inadequate facilities for exception handling. (Grudin, 1990; Grudin, 1994)

[3]See Thimbleby *et al*, (1994) for a discussion of 'trust' in CSCW.

[4]Whether this will always be the case is discussed in Hollan and Stornetta (1992), and is the subject of much futuristic virtual reality research.

These issues are inherent in collaboration and its mediation. When computers are used to support group work there is a further imposition of effort. These sources of effort are investigated in the following sections.

The effort of system requirements

Many groupware systems explicitly require additional effort from users in order to support their functionality (Cockburn and Thimbleby, 1993). Usually this effort takes the form of structured information (termed *guidance*). Guidance-dependence is best exemplified by the wide range of applications that support and enhance asynchronous messaging through the use of semistructured message templates (Lee and Malone, 1990; Malone *et al.*, 1992) that must be selected and filled-in by users. If guidance is initially absent or incorrect then, for successful operation, a third party user must provide or correct it. Without explicit guidance such systems render false information: potentially more damaging than no information (Cockburn, 1993).

Effort imposed by lacking flexibility

Several CSCW applications have been based on explicit theories of co-operative tasks: examples include speech–act theory (Flores *et al.*, 1988), IBIS (Conklin and Begeman, 1988), or the predetermined roles and tasks in Quilt (Leland *et al.*, 1988). These systems intentionally constrain the bounds of co-operative work.

Inflexible and constraining systems are unlikely to be popular. They enforce a form of 'work to rule': a phrase synonymous with inefficient, restricted and *inflexible* working practices. Although users may find ways to work around system-imposed restrictions (perhaps using paper notes to record personal views), such strategies illuminate system inadequacies.

Effort imposed by lacking integration

Sources of groupware effort go beyond the requirements and flexibility constraints of each *independent* system. Effort is also required to manage the various tools, facilities and communication mechanisms that make up the work environment.

Personal work requires people to make *transitions* (changes in their styles and methods of working) between their support tools. In *group* work, additional transitions are required: between single- and multi-user applications, and between alternative communication mechanisms. With infrequent use, the effort of relearning the interface may discourage participation altogether.

Inadequate integration between groupware and other computer tools is not only a source of user-effort, it is also a missed opportunity. Computers can integrate access to a variety of information about communications and collaborators, they can actively initiate collaborations, and can carry out autonomous processing to establish suitable colleagues or communicants (Cockburn and Greenberg, 1993).

Adoption and critical mass

Although the promise of work enhancement may encourage use, groupware tools are prone to a vicious circle that restricts the realisation of system borne

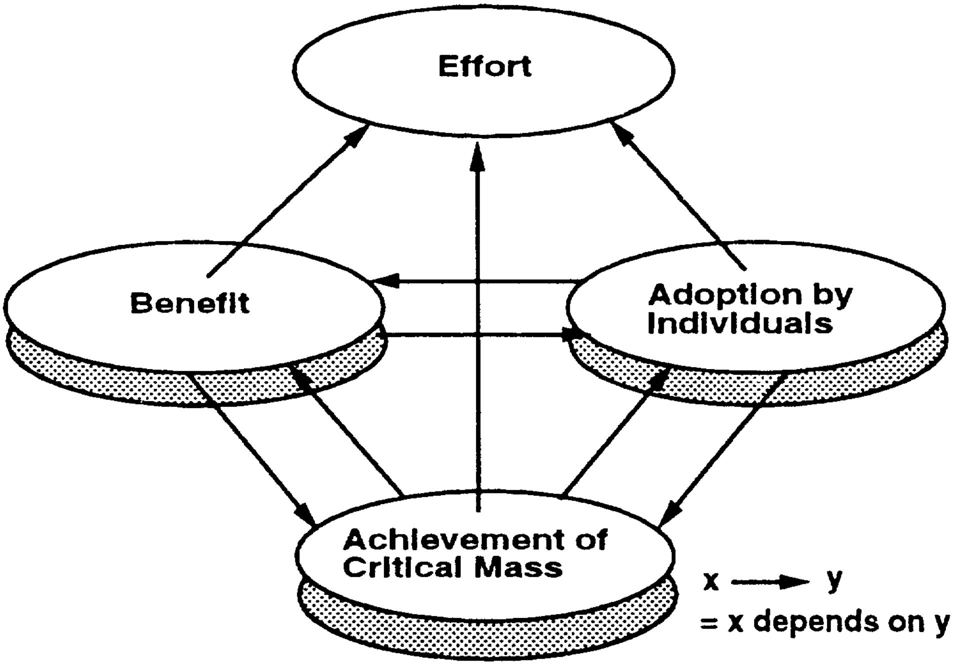

Figure 1. 'Vicious circle' of dependencies in groupware adoption

work enhancements (see Figure 1). The key determinant in this vicious circle is the level of *effort*.

- **Benefit and benefit-lag.** Willingness to adopt a system is dependent on the benefits derived from its use, and during adoption this is primarily determined by *immediate* gains. All computer systems, however, suffer from 'benefit-lag', the period during which the effort put into mastering a system out-weighs the benefit received. In the design of Capture Lab, Mantei (1989) notes that: ". . . a high learning threshold would cause meeting participants to reject the technology".
- **Attainment of critical mass.** Achieving critical mass depends on adoption by a *sufficient* group of individuals. Sufficiency in this context is contingent on the group, individual, and task requirements: in one group-task the main factor for example might be the number of collaborators, and in another, the involvement of particular individuals might be the main determinant.
- **Adoption by individuals.** Personal use of systems is encouraged if the rewards for doing so are clearly apparent: personal use is most likely to be stimulated by personal benefits.
- **The vicious circle of adoption.** Critical mass depends on adoption by individuals which is encouraged by benefits, but the benefits are contingent on a critical mass of users. All these properties must be simultaneously available before groupware can become successful. This situation appears to foretell a gloomy future for groupware!

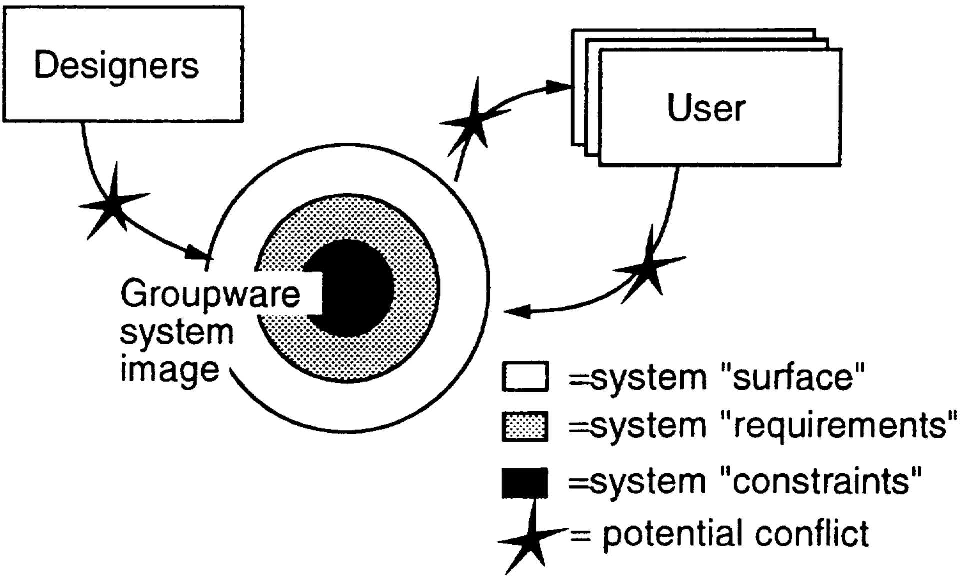

Figure 2. Generic 'layers' of groupware's system image

What is required is some sort of kick-start: a break in the vicious circle. The dominant role of effort throughout the system use must also be minimised. The groupware design principles, described in the following sections, offer generic strategies that work towards these goals.

From groupware problems to principles

Thus far, the problems encountered by groupware have been examined from the user's perspective. In this section, to aid the development of design principles for groupware, the user's effort is attributed to a set of generic system properties. These properties collectively form a groupware *system image* (see Figure 2): the intermediary between the designer's *conceptual model* of the system, and the user's *mental model* of the system (Norman, 1988). Conflicts between the system image and the user's mental model were used by Norman to explain user problems in the articulation of tasks and the comprehension of feedback in single-user software (Norman, 1983; Norman, 1988).

The HCI issues addressed by Norman's system image remain present in groupware, and are depicted in Figure 2 in the *system surface* layer of the groupware system image. In addition to direct interface considerations, groupware impinges on users in two additional (generic) ways: the additional work demanded to achieve its functionality, shown as the *system requirements* layer; and the constraints imposed on the users' way of working, shown as the *system constraints* layer.

The final observation in mapping from groupware problems to principles is that no single groupware tool is a holistic environment for collaboration. Differing artifacts and tools (which may or may not be computerised) satisfy specific tasks and personal tastes to varying degrees. The observation that designers must consider integration between tools would seem redundant were it not for the breadth of research that specifically addresses these issues — for example, (Ishii *et al.*, 1992; Malone *et al.*, 1992; Ohukubu and Ishii, 1990; Perlman, 1992; Press, 1992).

The groupware design principles described in the remaining sections of the paper each address a layer of the groupware system image, including issues external to the system.

Maximise personal acceptance

Maximising personal acceptance is concerned with encouraging individuals to adopt new systems. There is a similarity in how users view tools for personal and group work (for example, a word processor and a collaborative writing system). A common question users ask about both types of tool is: "What can it do for me?" During initial system use, this question will carry an additional component, "now". Although frequently the casualty of misplaced design attention, user interface issues are critical in groupware.

Beyond strict adherence to user-interface guidelines, some strategies for encouraging personal acceptance of groupware are described below.

Catchpenny systems
Feature ticking is a sales ploy used to add instant appeal to a wide range of modern products. Attractive features and additional facilities supplement the core functionality, turning attention to fancy bells and whistles. Although not condoning the design of poor (but feature rich) groupware, a form of feature ticking could be used to supply instant user-appeal (Cockburn, 1993).

The "Reflexive Perspective"
of CSCW blurs the distinction between support mechanisms for personal and group work (Thimbleby *et al.*, 1990). It is common for people to work on several machines (one at the office, one at home, a lap-top, and an assistant's machine) and several projects may be pursued at different times. Personal group-like behaviour is further illustrated when our separate work roles are examined (Cockburn and Thimbleby, 1991). These roles include the following: a *management* role in which decisions about work co-ordination are made; a *worker* role in which the actions necessary to advance or complete the work are executed; a *meta-management* role in which personal assistants (human or computer) are instructed about appropriate actions.

With multiple tasks, roles, and work places, the individual's co-ordination requirements are similar to those of asynchronously collaborating co-workers. By exploiting the similarities, people benefit from the consistent interface to the personal and collaborative work environments. Skills transfer from one environment to the other, and the effort of learning and remembering separate interfaces are shared over a wider range of tasks.

Champions and encouragement
CSCW evaluations (Ehrlich, 1987; Fafchamps *et al.*, 1991; Francik *et al.*, 1991)

have shown that groupware enthusiam is greatly enhanced by 'champions' or 'evangelists'. These people are deployed during initial system use to promote the use of the technology, raise awareness of what it can achieve, and generally encourage system use.

Minimise requirements

User effort plays a pivotal role in groupware adoption (See Figure 1). The intention of minimising requirements is to reduce the disparity between groupware's costs and benefits to user-acceptable levels. Strategies for achieving this goal are summarised below[5].

Avoid dependence on user actions
Systems that depend on users providing structured information (guidance) encounter serious problems when the guidance is absent. Rather than depend on guidance, a more acceptable approach is to use it when available, but maintain correct operation when it is absent. It has been argued that a relaxed approach of this nature is impractical:

> "Can a CSCW application succeed if doing the extra work is left to individual discretion? Unfortunately, probably not." (Grudin, 1988, p. 86.)

The major problem for system designers, then, is *how* to retrieve guidance. The user is potentially the most accessible and accurate resource, but research and experience has shown that systems can profitably look elsewhere (Cockburn and Thimbleby, 1993; Kozierok and Maes, 1993; Maes, 1994).

Use what's available 'for free'
Although guidance is required to provide certain facilities, there are sources other than the users. Information is accessible to computers through the process of communication: for instance, e-mail messages contain header information that includes the who, when, where, and other information. Existing 'for free' approaches are wide ranging and include the following: the use of e-mail headers to infer conversational relationships between e-mail messages (Cockburn and Thimbleby, 1993); natural language parsing and keyword scans within text-based communication (Shepherd *et al.*, 1990); adaptive and learning systems that modify their performance dependent on user characteristics (Kozierok and Maes, 1993; Maes, 1994); Latent Semantic Indexing (Foltz and Dumais, 1993) which infers the semantic distance between text documents.

Enable shifts of cost and benefit
Designers and manages who strive for efficiency-enhancing groupware have, typically, assumed that people are willing to work for the benefit of others (Grudin, 1988; Nagasundaram, 1990). This assumption ignores social effects

[5]For a complete discussion of the strategies used to implement minimised requirements, and point systems demonstrating their use, see Cockburn (1993) and Jones (1994).

including the users' reluctance (or inability) to carry out actions that provide no *personal* benefit.

By shifting the provision of guidance (the cost) onto users gaining the benefit, the cost/benefit disparity is reduced — users execute additional actions when they are willing and able. Tapestry (Goldberg *et al.*, 1992) underlines such an approach: through 'collaborative information filtering' it exploits those people who are willing to carry out additional work altruistically.

The applicability of a cost-shifting approach depends on the politics and hierarchical structure of the organisation in which it is implemented. Although it may be reasonable to expect subordinates to work on behalf of a manager, the reverse may not be true. Social protocols should be allowed to resolve conflicts between expectations of actions at one level and execution of actions at another.

Minimise constraints

Minimising requirements is concerned with the implementation stage of groupware development. It focuses on how systems retrieve the information they require. Minimising constraints attends to problems arising at earlier, more abstract, stages of system development. It examines the models and theories underlying groupware. The aim is to avoid inflexible and constraining styles of use.

In theory, rigid working practices can support highly efficient organisations. In practice, few organisations operate according to such deterministic methods; furthermore, they cannot be made to do so (Nagasundaram, 1990; Suchman, 1987). This principle for minimised constraints argues for groupware that leaves the users free to develop protocols governing collaborative work as they, rather than their systems, see fit.

Strategies for achieving minimal constraints, summarised below, primarily aim to increase designers' awareness of problems arising from inflexible and rigid systems. Specific and detailed strategies are likely to be inappropriate due to the diversity of the models implemented by groupware.

Be aware of the two-level perspective of technology

Sproull and Kiesler (1991) examined the conflict between the increased efficiency enabled by computer support and the negative social implications. They categorised the effects of technology under two 'levels of perspective'. The distinction between these levels can be expressed by the questions "what is *possible* with technology?" at the first level, and "how will it be used?" at the second.

Groupware designers, and all those involved in system development, must be aware of the social implications inherent in group work support. Technology capable of enhancing organisational efficiency will fail if social factors are ignored. Design alterations based on projections of a system's social implications may temper the efficiency improvements achievable, but it is better to provide acceptable mechanisms providing some benefit than unacceptable ones that, despite great potential, fulfil none.

Beware of rigid models and theories.
CSCW research into collaborative activity promises to yield workable models of collaborative activity in the future. The lack of maturity and incomplete state of this research, however, makes the use of "universally applicable" explicit models in current groupware inappropriate. The dangers of overly constraining the working practices of groupware users are widely cited:

- *on the Coordinator* (Flores *et al.*, 1988; Winograd, 1987)

  "... the strength of its theoretical foundation also appears to be its Achilles heel" (McCarthy and Monk, 1994, p. 51).
  "The conversational templates appeared to be more a straight-jacket than a communication medium.' (Carasik and Grantham, 1988, p. 64).

- *on Cosmos* (Dollimor and Wilbur, 1991)

  "The apparent inflexibility of the Cosmos system may be partly due to the use of a particularly rigid conception of role" (Jirotka *et al.*, 1992, p. 98).

- *in general*

  "Technological protocols can be overly restrictive: a group's idiosyncratic working style may not be supported, and the system can constrain a group that needs to use different processes for different activities." (Ellis *et al.*, 1991, p. 51).
  "... reproduce[ing] the formal lines of control and communication may even hinder cooperation" (Jirotka *et al.*, 1992, p. 99).
  "Nice technology, but it doesn't allow us the flexibility to handle the many exceptions." (Ellis and Wainer, 1994, p. 76).

Constraints on working practices are not always a result of explicit models of work processes. Implicit assumptions about collaborative work can be easily made and embedded into systems. Such is the case in the Cognoter subsystem of Colab (Stefik *et al.*, 1987; Stefik *et al.*, 1988):

> "[Cognoter] slipped up on implicit aspects of the system, places where the designers didn't realise they were making choices" (Tatar *et al.*, 1991, p. 207).

This warning on the use of deterministic models of work processes is not an attempt to discredit the on-going research into workflow systems. Current workflow research directly addresses the problems of inflexibility and over-structuring (Ellis and Wainer, 1994; Swenson *et al.*, 1994) through support for dynamic tailoring (Trigg and Bodker, 1994). The interface to the workflow tailoring facilities will be critical. It remains to be seen whether everyday users will embrace the powerful functionality offered by these systems.

Open, unconstrained enhancement
While models and theories of collaborative activity are under development, open and unconstrained systems allow users to develop protocols as they see fit.

User-specific models can be used to supplement an open system, but they should not impose constraints on collaborative tasks or on their mediation.

Existing groupware applications exemplify various 'open' approaches. Toolkits such as GroupKit (Rosemand and Greenberg, 1992) support a range of open protocols for floor control and other facilities. *Milo* (Jones, 1992) avoids modelling specific writing styles/roles in order to free co-authors from constraints. Summarising the aims of the 'minimise constraints' principle, the *Amsterdam Conversation Environment* had as its central design philosophy the notion of support through "non-dependency creating enablement" (Dykstra and Carasik, 1991).

External integration

The first three groupware principles are primarily concerned with design and use of groupware *in isolation*. In contrast, maximising external integration requires designers to consider their system's role within, and relationship to, the entire work environment. In this extended collaborative context group members use competing systems to execute similar tasks, and a variety of tools (computer and non-computer based) are drawn on to support and assist collaboration.

Enabling external integration attends to the user-effort that results from the changes (transition, also termed "seams," and "barriers" (Baecker, 1993)) made between differing applications or communication environments. Its primary aims are twofold:

- **Curative** — to reduce the number and magnitude of transitions between tools and facilities employed in collaborative work.
- **Augmentative** — to improve and integrate access to resources that serve communication and collaboration requirements.

This section describes groupware approaches that reduce transitions in CSCW.

Video fusion

Best exemplified by *ClearBoard* (Ishii *et al.*, 1992) video fusion systems enhance compatibility between personally favoured tools. Fusion allows separate video images (perhaps a computer screen, and a human face) to be overlayed like layers of transparent acetate. In this way two (or more) otherwise incompatible applications, or work environments, can be used together.

Heterogeneous environments

These augment collaboration by drawing together access to, and information about, collaboration resources. They work towards an "integrated portfolio of media" (Bair, 1989). Telefreek (Cockburn and Greenberg, 1993) provides an extensible CSCW environment based on, but not limited to, standard networked computers. By drawing together information sources, communication mechanisms, and collaboration applications, Telefreek users are provided with a platform for communication and collaboration.

Minimise dependence on structure and format
System specific information formats and structures reduce the potential for integration. Minimally, groupware that is intended for general release (rather than research point systems) must follow relevant standards. Many standards allow flexiblity and additional structure to be added within their specifications, but designers must consider the impact of such structures on colleagues who do not have access to the same structuring mechanisms (Lee and Malone, 1990). Systems built on existing communication media should do so monotonically: new facilities and enhanced features should not affect those already in use.

Implementation platforms
Incompatibilities between hardware and interface platforms are a primary concern for groupware developers. To avoid these problems designers must either replicate some of the implementation to support a variety of hardware, choose a suitable hardware-independent development platform (such as the X-Window system (Scheifer and Gettys, 1986)), or use an interface development application that can generate code for several graphical user interface environments.

Cross-platform interface generators are becoming available and are likely to substantially ease the implementation of integrable groupware. Groupware development toolkits, such as GROUPKIT (Rosemand and Greenberg, 1992) and OVAL (Malone *et al.*, 1992) also promise to increase the integrability of systems.

Discussion and future work

The main function of the groupware design principle proposed is to provide an extendable framework for design strategies. An abstract principle such as 'minimise requirements' may be conceptually correct but is of little practical use to designers of 'real world' applications. Consequently, for each of our principles we suggest strategies that can be adopted to achieve the principle's aims.

In fact, our guidance framework for groupware designers exists at three levels: principles, strategies and techniques. Principles are critical high level design aims, and several strategies may be adopted to adhere to a principle. For example, one strategy that will minimise system requirements of users is 'use what is available for free'. This strategy may be employed in various ways, dependent on the application area, application environment, and so on. Techniques are the low-level guidance to a good design which take such issues into account, representing principles and strategies in the system itself. Our suggestion that e-mail headers can be examined to infer conversational relationships is one such technique.

We currently concentrate on principles and strategies. To be adopted (by designers) these principles and strategies, like groupware applications, must be acceptable to their users. Therefore it is critical that the strategies and techniques suggested here are further expanded and refined. Additionally designers must be guided in the selection of strategies and techniques that are appropriate to their particular problem area. One issue that we highlight is that of the negative effects of systems increasing or requiring user effort. Adoption of our principles, however, will likely increase the amount of effort designers invest in systems: a theme of the principles is a shift of effort in supporting the interaction from user to systems. Therefore systems are likely to be larger and more complex. One possible implication is that some designers may view adherence to the principles as too time-consuming, too difficult, uneconomical and so on. We do not present our principles as a 'take it or leave it' whole. Designers may select those that are appropriate to their requirements and achievable within the constraints imposed upon them.

The appropriateness of the principles and strategies is a critical issue, likely to be decided by designers at the point of design. We do not advocate prioritization of the principles or strategies. Each can benefit users of groupware in different ways, and each is of equal importance. Prioritization is also inappropriate because different principles will clearly be dominant in different contexts. For example, external integration is dominant in asynchronous distributed groupware: personal acceptance is critical in a context such as collaborative writing where much of the document production may be done on an individual basis. Part of the development of this work will be to consider the relationships between the principles in different collaborative contexts.

It is too early to address fully the utility of the principles for several reasons. First, it is difficult to evaluate groupware itself, and techniques for doing so are not yet well developed. In time, comparisons between systems that adhere to the principles and those that do not will be required. Evidently the systems will have to be suitably comparable. Additionally the evaluation techniques adopted will have to be sufficiently sensitive to differentiate between improvements and perhaps degradation in system design that are directly attributable to the principles and not the multitude of other variables present in groupware usage. Second, there is not a currently large base of applications that have been designed according to the principles. Hence the other important indicator of utility of the principles — the experience of designers — is currently limited.

We can, however, report that three systems have been developed using the principles. *Mona* is a conversation-based electronic mail platform (Cockburn, 1993; Cockburn and Thimbleby, 1993). MILO is a tool to support asynchronous, distributed collaborative writing tasks (Jones, 1992; Jones, 1994). TELEFREEK (Cockburn, 1993; Cockburn and Greenberg, 1993) support an extendable and customisable platform for communication and collaboration resources. An objective analysis of the process and products is difficult, but the principles seemed to ease the task, helping to answer questions about design, and the systems have been successfully used in an integrated way in a collaborative writing task. More widespread use of the principles is required for our observations to be verified. This will serve two purposes. First, a wider, more objective consideration of the principles will be possible. Second, further strategies and techniques will emerge and will be integrated into the existing design framework.

We see the potential for wider utilisation of the principles than has already been presented. First, the principles may be used for evaluation of groupware in a way similar to the use of user interface design principles in techniques such as heuristic evaluation (Nielsen, 1992; Nielsen, 1993). Second, the use of the principles and some of the strategies may be extensible beyond groupware to other interactive systems, as they address issues in the personal context of CSCW systems. Further strategies and techniques may be developed within the framework which do not necessarily relate to collaborative systems.

Conclusions

In this paper we have provided a cross-disciplinary perspective on groupware development that focuses on those most greatly affected by its deployment: the users. As an early step forward in the development of best practice in groupware design, we have noted the common failings of groupware, and from those observations we have derived a set of design principles that encapsulate the problems and ways to avoid them.

References

Baecker, R.M. (1992) 'The future of groupware for CSCW' in **Baecker, R.M. (ed)** *Groupware and Computer-Supported Cooperative Work* Morgan Kaufmann, 851–853

Bair, J.H. (1989) 'Supporting cooperative work teams with computers: addressing meeting mania' *Proc. 34th IEEE Computer Society Int. Conf.* IEEE Press, 208–217

Carasik, R.P. and Grantham, C.E. (1988) 'A case study of CSCW in a dispersed organisation' *Proc. CHI'88 Conf. Human Factors in Computing Systems* ACM Press, 61–65

Cockburn, A.J.G. (1993) 'Groupware design: principles, prototypes, and systems' *PhD Thesis*, University of Stirling, Scotland

Cockburn, A.J.G. and Greenberg, S. (1993) 'Making contact: getting the group communicating with groupware' in *Proc. COOCS'93: Conf. Organisational Computing Systems* (Milpitas, CA, USA)

Cockburn, A.J.G. and Thimbleby, H.W. (1991) 'A reflexive perspective of CSCW' *ACM SIGCHI Bull.* **23**, 3, 63–68

Cockburn, A.J.G. and Thimbleby, H.W. (1993) 'Reducing user effort in collaboration support' *Proc. OZCHI'92* (Bond University, Australia), 142–149

Conklin, J. and Begeman, M.L. (1988) 'gIBIS: a hypertext tool for exploratory policy discussion' *ACM Trans. Office Info. Sys.* **6**, 4, 303–331

Dollimore, J. and Wilbur, S. (1991) 'Experiences in building a configurable CSCW system' in **Bowers, J.M. and Benford, S.D. (eds)** *Studies in Computer Supported Cooperative Work: Theory, Practice and Design* North-Holland, 173–181

Dykstra, E.A. and Carasik, R.P. (1991) 'Structure and support in cooperative environments: the Amsterdam Conversation Environment' *Int. J. Man-Mach. Stud.* **34**, 419–434

Ehrlich, S.F. (1987) 'Strategies for encouraging successful adoption of office communication systems' *ACM Trans. Office Info. Syst.* **5**, 4, 340–357

Ellis, C.A., Gibbs, S.J. and Rein, G.L. (1991) 'Groupware: some issues and experiences' *Commun. ACM* **34**, 1, 38–58

Ellis, C.A. and Wainer, J. (1994) 'Goal-based models of collaboration. *Collaborative computing* **1**, 1, 61–86

Fafchamps, D., Reynolds, D. and Kuchinsky, A. (1991) 'The dynamics of small group decision-making using electronic mail' **Bowers, J.M. and Benford, S.D. (eds)** *Studies in CSCW: Theory, Practice and Design* North-Holland, 211–224

Flores, F., Graves, M., Hartfield, B. and Winograd, T. (1988) 'Computer systems and the design of organisational interaction' *ACM Trans. Office Info. Syst.* **6**, 2, 153–172

Foltz, P.W. and Dumais, S.T. (1993) 'Personalized information delivery: an analysis of information filtering methods' *Commun. ACM* **35**, 12, 51–60

Francik, E., Rudman, S.E., Cooper, D. and Levine, S. (1991) 'Putting innovation to work: adoption strategies for multimedia communication systems' *Commun. ACM* **34**, 12, 53–63

Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. (1992) 'Using collaborative filtering to weave an information tapestry' *Commun. ACM* **35**, 12, 61–70

Grudin, J. (1988) 'Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces' *Proc. CSCW'88* 85–93 (Portland, OR, USA)

Grudin, J. (1990) 'Groupware and cooperative work: problems and prospects' in **Laurel, B. (ed)** *The Art of Human–Computer Interface Design* Addison-Wesley

Grudin, J. (1994) 'Groupware and social dynamics: eight challenges for developers' *Commun. ACM* **37**, 1, 93–105

Hollan, J. and Stornetta, S. (1992) 'Beyond being there' *Proc. CHI'92: Conf. Human Factors in Computing Systems* Addison-Wesley, 119–125

Ishii, H., Kobayashi, M. and Grudin, J. (1992) 'Integration of inter-personal space and shared workspace: Clearboard design and experiments' in *Proc. 1992 ACM Conf. Computer Supported Cooperative Work* ACM Press, 33–42

Jirotka, M., Gilbert, N. and Luff, P. (1992) 'On the social organisation of organisations' *Computer Supported Cooperative Work: Int. J.* **1**, 1–2, 95–118

Jones, S. (1992) 'MILO: a computer based tool for (co)authoring structured documents' in **Sharples, M. (ed)** *Computer Supported Collaborative Writing* Springer-Verlag

Jones, S. (1994) 'Easing the writing task: designing computer based systems to help authors' *PhD Thesis* Department of Computing Science and Mathematics, University of Stirling, Scotland

Kozierok, R. and Maes, P. (1993) 'A learning interface agent for scheduling meetings' *Proc. 1993 Workshop on Intelligent User Interfaces* ACM Press, 81–88

Lee, J. and Malone, T.M. (1990) 'Partially shared views: a scheme for communicating among groups that use different type hierarchies' *ACM Trans. Office Infor. Syst.* **8**, 1, 1–26

Leland, M.D.P., Fish, R.S. and Kraut, R.E. (1988) 'Collaborative document production using Quilt' in *Proc. 2nd Conf. Computer Supported Cooperative Work* ACM Press, 206–215

Maes, P. (1994) 'Agents that reduce work and information overload' *Commun. ACM* **37**, 7, 31–40

Malone, T.W., Lai., K.-Y. and Fry, C. (1992) 'Experiments with OVAL: a radically tailored tool for cooperative work' in *Proc. 1992 ACM Conf. Computer Supported Cooperative Work* ACM Press, 289–297

Mantei, M. (1989) 'Observation of executives using a computer supported meeting environment' *Decision Support Systems* **5**, 153–166

McCarthy, J.C. and Monk, A.F. (1994) 'Channels, conversation, cooperation, and relevance: all you wanted to know about communication but were afraid to ask' *Collaborative Computing* **1**, 1, 35–60

Muller, M.J. and Kuhn, S. (1993) 'Introduction to the special issue on participatory design' *Commun. ACM* **36**, 4, 24–28

Nagasundaram, M. (1990) 'Style and substance in communication: Implications for message structuring systems' *ACM SIGOIS Bull.* **11**, 4, 33–41

Nielsen, J. (1992) 'Finding usability problems through heuristic evaluation' *Proc. CHI'92 Conf. Human Factors in Computing Systems* Addison-Wesley, 373–380

Nielsen, J. (1993) *Usability Engineering* Academic Press

Norman, D.A. (1983) 'Design principles for human–computer interfaces' *Proc. CHI'83* ACM Press, 1–10

Norman, D.A. (1988) *The Psychology Of Everyday Things* Basic Books

Ohukubu, M. and Ishii, H. (1990) 'Design and implementation of a shared workspace by integrating individual workspaces' in *Lochovsky, F.H. and Allen, R.B. (eds) Proc. Conf. Office Info. Syst.* ACM Press, 142–146

Perlman, G. (1992) 'A vision of universal functionality for tomorrow's user interfaces' in **Rees, M.J. and Iannella, R. (eds)** *Interface Technology: Advancing Human–Computer Communication. Proc. OZCHI 1992* (Bond University Australia, Queensland) 1–14

Press, L. (1992) 'Collective dynabases' *Commun. ACM* **35**, 6, 26–32

Roseman, M. and Greenberg, S. (1992) 'Groupkit: a groupware toolkit for building real-time conferencing applications' in *Proc. 1992 ACM Conf. Computer Supported Cooperative Work* ACM Press, 43–50

Scheifler, R.W. and Gettys, J. (1986) 'The X-Window System' *ACM Trans. Graph.* **5**, 2, 79–109

Shepherd, A., Mayer, N. and Kuchinsky, A. (1990) 'Strudel: an extensible electronic conversation toolkit' *Proc. 3rd Conf. Computer Supported Cooperative Work* 93–104

Sproull, L. and Keisler, S. (1991) *Connections: New Ways Of Working In The Networked Organization* MIT Press

Stefik, M., Bobrow, D.G., Foster, G., Lanning, S. and Tatar, D. (1987) 'WYSIWIS revised: early experiences with multiuser interfaces' *ACM Trans. Office Information Systems* **5**, 2, 147–167

Stefik, M., Foster, G., Kahn, K., Bobrow, D.G., Lanning, S. and Suchman, L. (1988) 'Beyond the chalkboard: computer support for collaboration and problem solving in meetings' in **Greif, I. (ed)** *Computer Supported Cooperative Work: A Book of Readings.* Morgan Kaufman, 334–366

Suchman, L. (1987) *Plans and Situated Actions: The Problems Of Human–Machine Communication* Cambridge University Press

Swenson, K.D., Maxwell, R.J., Matsumoto, T., Saghari, B. and Irwin, K. (1994) 'A business process environment supporting collaborative planning' *Collaborative Computing* **1**, 1, 15–34

Tatar, D.G., Foster, G. and Bobrow, D.G. (1991) 'Designing for conversation: lessons from Cognoter' *Int. J. Man–Mach. Stud.* **34**, 2, 185–209

Thimbleby, H.W., Anderson, S. and Witten, I. (1990) 'Reflexive CSCW: supporting long-term personal work' *Interacting with Computers* **2**, 3, 330–336

Thimbleby, H.W., Marsh, S.P., Jones, S.R. and Cockburn, A.G. (1994) 'Trust in CSCW' in **Scrivener, S. (ed)** *Computer-Supported Cooperative Work* Ashgate Publishing, 253–271

Trigg, R.H. and Bodker, S. (1994) 'From implementation to design: tailoring the emergence of systemization in CSCW' in *Proc. 1994 ACM Conf. Computer Supported Cooperative Work* ACM Press, 45–54

Winograd, T. (1987) 'A language/action perspective on the design of cooperative work' *Human–Computer Interaction* **3**, 1, 3–30