Sociologists can be Surprisingly Useful in Interactive Systems Design

Ian Sommerville, Tom Rodden, Pete Sawyer and Richard Bentley,

Computing Department, Lancaster University, Lancaster, LA1 4YR, UK
Tel: +44-524 65201
Fax: +44-524 381707
EMail: is@comp.lancs.ac.uk

This paper makes a case, to system developers, for inter-disciplinary working and the involvement of sociologists in the systems design process. Our argument is based on the fact that effective systems must take account of the social context in which these systems are situated. The paper is based on our experiences of working with sociologists in a study of air traffic control automation. We describe the model of working which we use and which we believe allows effective utilisation of the skills of both disciplines. We then set out pre-cursors for effective inter-disciplinary collaboration and how people from radically different backgrounds can work in harmony. Finally, we discuss some of the problems of collaboration which are likely to arise.

Keywords: systems design, ethnography, inter-disciplinary working

1. Introduction

This paper discusses some of the problems and benefits of collaboration between sociologists and computer scientists in the interactive systems design process. It is based on our experiences of developing a proposal for and carrying out a research project concerned with investigating database display mechanisms for air traffic controllers.

This project is concerned with investigating requirements for an automated display system for UK air traffic controllers. Currently, aircraft descriptors are printed from the database onto paper strips before they enter controlled airspace. These strips are mounted in holders and placed in racks of about 20 strips. The collection of strips presents controllers with an overall picture of the sector of the airspace they are controlling.

The task of air traffic control is inherently collaborative with controllers constantly monitoring each other's work and negotiating with controllers of adjacent sectors as to the most suitable height and flight path for particular aircraft. A very complex and subtle way of working has evolved which is inherently safe and controllers are reluctant to accept any

automated system which does not take the implicit safety characteristics of the manual system into account. Previous attempts to automate the aircraft descriptor display system have been rejected by controllers.

Our work is distinct from these previous automation projects inasmuch as we are basing the system requirements on an ethnographic study which is being carried out by sociologists. This study involves detailed observation of the controllers at work as well as in-depth interviews with them about their job and how they have evolved successful working practices. The sociologists will also be involved in the system evaluation process, observing how controllers modify their working practices to adapt to the new, automated system.

We are computer scientists and software engineers concerned with developing interactive systems so the views presented here are inevitably biased by our own perspective. Our sociologist collaborators would undoubtedly write a different paper although we hope that they would not disagree too strongly with the opinions here. We make no apologies for presenting a one-sided case as we are addressing this paper at the interactive systems development community. As discussed below, we believe that this community can benefit significantly by including sociologists in the design process.

The role of sociologists has, of course, already been recognised in CSCW and sociologists have provided important insights in this area. Pioneering work by Suchman (1983) has been followed by later studies (Gerson and Star, 1986; Harper *et al.,* 1991*;* Heath and Luff, 1991). However, we believe that the contribution of sociologists is not confined to CSCW systems but is a more general one which is applicable in most application studies. While this body of previous work has exposed the complexities of the interaction between people and automated systems, our work is unusual in that an explicit task of the sociologists is to help derive the software system requirements.

The particular models and mechanisms which we have adopted in this project are discussed elsewhere (Bentley *et al.,* 1992). We believe that our experiences of collaboration with sociologists in this project have been mutually beneficial and that more and more such collaborations will take place across the HCI community. This paper, therefore, reports on some of our collaboration experiences. We have learned from these experiences and, in the remainder of the paper, we suggest how effective collaboration can be organised and potential problems which can arise.

Existing computer applications have been immensely successful and have radically transformed our society. The present standard of living enjoyed in Western Europe and North America could not have been achieved without effective application systems. By and large, these systems were developed without the aid of sociologists. Why then is there now a need for sociologists to be involved in the design process?

We argue that computerisation up till now has concentrated on relatively simple applications with large economic payoff. Corporate computer applications automated existing manual systems giving large productivity increases and the potential for introducing new innovative systems. Personal computer applications such as word processors and spreadsheets supported high productivity increases for personal work across

a wide range of disciplines. Until relatively recently, few of these corporate or personal application systems were easy to use. Users had to invest a great deal of time and effort in learning to use the system. However, the potential benefits were so great and so obvious that many users decided that it was worth spending this time in learning to use a computer system.

Adapting to the new system usually meant that users had to make significant changes in working practice. For example, word processors mean that many more people key input directly rather than writing it on paper then passing it to a typist. To a large extent, system developers have not needed to concern themselves about the subtleties of working practice; they assume that users adapt to the automated systems. Hence, many if not most current systems have significant usability problems.

Current application systems have been successful in spite of their usability problems because they offered so much. An inevitable consequence of the law of diminishing returns is that the next generation of application systems will offer a lower productivity improvement; users will be unwilling to change their working practices to adapt to these systems because the advantages from that change will not be obvious. Hence systems have to be more usable in order to be accepted.

An essential characteristic of usability is conformance to existing working practice. Users will not change the way they work to adapt to a computer system if the benefits are not significant and obvious. We must therefore have a clear understanding of the workplace and the way in which humans interact with each other in that workplace. We must also understand how they actually use interactive systems and the ways in which they manage and process information.

While the way in which many jobs should be carried out is prescribed in company handbooks, in reality the individuals doing these jobs evolve their own ways of working and interaction. Indeed, labour disputes sometimes involve 'working-to-rule'. These are effective because following the rules means that the overall productivity of workers is significantly diminished.

The development of techniques of user-centred design (Kyng, 1988; Norman and Draper, 1986) has recognised the importance of involving users, who have a detailed understanding of their job, in the design process for interactive systems. In user-centred design, end-users become part of the design team and are continually available to comment on and to evaluate proposed designs. They may themselves suggest appropriate user interface designs.

Involving end-users in the design process is essential but we argue that *user-centred* design on its own is not enough. Rather, the design process must also take account of the *social context* where the computer system is installed. While users understand the details of their own task, their attention is typically on these tasks rather than how they fit into an overall organisation of working activities. Furthermore, different users evolve different ways and shortcuts of carrying out the same task.

There is a need for someone to look at 'the big picture' and to study the overall functioning of an organisation. It is this role the sociologist should adopt. We are not suggesting here

that the sociologist can completely replace the user and that this approach should supplant participative design. Rather, we believe that using observations of how users work brings a new perspective to the system which complements user advice, which may perhaps be provided through participative design.

We argue that sociologists are the best people to study and understand organisations because of the theories, methods and procedures which they have developed. Sociologists have extensive experience in studying societies in an objective way without prejudices as to what they might discover. Techniques such as ethnography (prolonged observation of a society) and conversational analysis have been developed by sociologists as ways of analysing social systems within complex organizations. Unlike technologists, they do not approach the study with preconceived notions of the application systems which might be developed.

2. Developing Systems with Sociologists

The role of the user interface has evolved over time to the point that computer systems and their interfaces are now soundly set within an organisational context. The motivation for computer scientists seeking input from other disciplines is a realisation that social and organisational knowledge is important for the success of future computer systems. The assumed role for sociology is in providing the missing feature from existing systems or as Grudin (1990) states:

"Since most work occurs in a social context, computers will support it more successfully if they implicitly or explicitly incorporate organisational or social knowledge"

One response to this realisation can be simply stated as "adding *factor X* will make existing computer systems into successful social systems". Within this model of working, sociologists provide the details of the how a group of individuals work together. The system, supporting group interaction, is then realised from this information by other system developers. Within this differentiation of disciplines, sociologists act as a form of system analyst. However, our initial investigation of the results produced by the examination of social systems (Harper *et al.,* 1991) convinced us that the process was not that simple.

The subtlety and complexity of even mundane social interactions cannot be captured by a 'systems analysis' phase. The sociologists involved in the project provide an understanding of what is really going on in the workplace. Because they establish a rapport with end-users and learn many of the subtleties of their job, they act as an active communication pathway between system development and the existing work practice of users.

Sociologists have to become the interface between the context of systems development and the work setting in which the system is placed. During the design process, individual design decisions may be analysed for their social effects. The sociologists may use their records of the ethnography or may go back into the work setting and look for particular characteristics which would be influenced by some design decision.

It is important for computer scientists to realise that their expectations in the analysis of organisations whilst appearing clear from their own perspective, place novel demands on

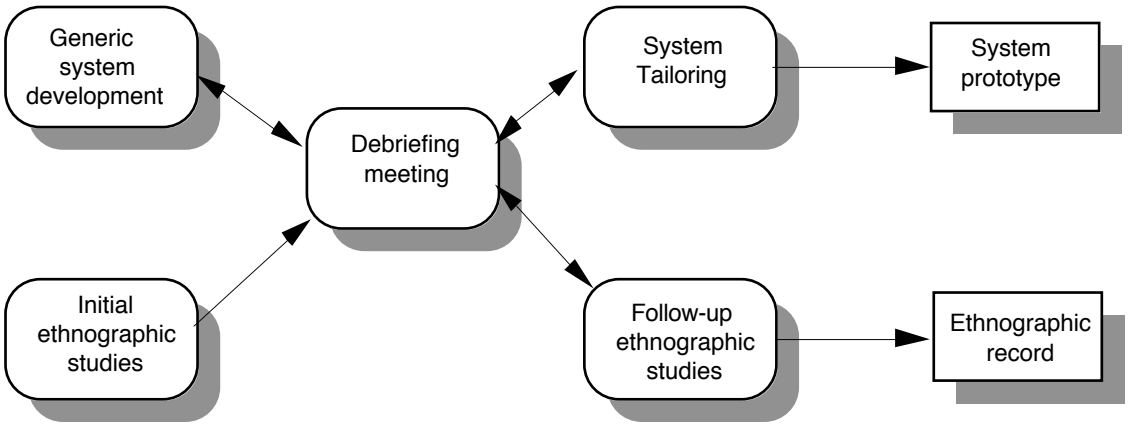

Figure 1 Integrating ethnography with system prototyping

sociologists. Social scientists are more used to asking questions and making observations rather than coming up with 'answers'. Social scientists such as Bannon and Schmidt (1991)working in the area of CSCW systems have suggested that a theoretical framework is needed to understand the interactions between the task requirements, the work organisation and the computer system. However, they freely admit that:

> "The required theoretical framework that would help analysts and designers to deal with these issues, however, is not imminent"

Given that we have few theoretical frameworks of any kind in the area of human-computer interaction, it is our belief that such a framework is unlikely to emerge for many years. It will be necessary to design systems which take the social organisation of work into account and pragmatic techniques to support system development with social science input must be developed.

2.1 System Development

Because of the complete lack of any existing theoretical framework for collaboration, we devised a development process which allowed system prototyping to take place in conjunction with on-going social studies. The model of development which we use in the air traffic control project and which we propose as a general model is shown in Figure 1.

As detailed system requirements cannot be predicted in advance, we believe that only a generator-based approach to prototype construction is likely to be successful. The first phase of software development must be to either find or build a generic system which can be easily tailored at a very detailed level. In parallel with this development, initial studies of the system of interest should be carried out. Regular (approximately monthly) debriefing meetings should be held where the sociologist describes the system. These influence the development of the prototype generator.

After the generator has been completed, the next stage is to develop a prototype system. This development is almost certain to raise a large number of questions about work

practices. These may be answered from the ethnographic record gathered in previous studies or may require further work by the sociologists with the explicit objective of finding answers to the developer's questions. This iterative process continues until a working prototype which may be exposed to end-users is available.

Input from the sociologists has two distinct forms in our approach to systems development, that of the *users' champion* and the *designer's conscience*. As the users' champion, the sociologists tells developers about the actual work practice and the users' perception of this work practice. This input is provided formally by the development of field notes which document the observations of the sociologists and less formally by the use of debriefing meetings where system developers question the ethnographer.

Our experience to date has suggested that these debriefing exercises are central to the understanding of the work process. During the meetings, systems developers often picked up the significance of particular work practices and questioned the sociologist for more details of them. These points were often aspects of the system which would pose particularly difficult problems of automation. More often than not, further ethnographic studies were required to resolve the issues raised by the systems developers.

In addition to insight into existing work practice, working with sociologists has fostered a heightened awareness of the impact of design decisions within our development team. This has resulted in the sociologists involved in the project acting as a designer's conscience. This effect is normally manifest as a series of questions to the sociologists of the form :

"If we decide to use X then users will not be able to carry out action Y. Is this OK or is Y a critical action?"

The sociologists involved in the project either provide a simple answer to this question or, more often, highlight a wider set of issues surrounding this question. This form of interaction requires close cooperation between both computer scientists and sociologists and involves a great deal of informal negotiation.

Our current practice which involves proposing a design then assessing it with our sociologist collaborators is very time-consuming. We perceive the need for some more structured means of analysing the ethnographers field notes so that they are more accessible to non-specialists. We want to be able to assess if sufficient information about the work context is available to allow us to be confident in our design decisions.

We have found it difficult to derive any systematic means of identifying key system requirements from the ethnographic record. We put four questions to the sociologists which we felt were important and which we hoped could be answered from their ethnographic studies:

1. What characteristics of the existing manual system are unimportant and need not be supported in an automated system?

2. What are important manual activities which need not be supported in an automated system because the activities are a consequence of the fact that no automated support is available?
3. What characteristics of the manual system must be replicated without change in an automated system?
4. What activities from the manual system may be supported in a way which is different from that used in the manual system?

On posing these questions, the immediate response of the sociologists was to question their validity and independence. A cynic might say that when you can't answer a question, you should attack its validity but we believe that the response we received illustrate a fundamental difference in ways of thinking.

In carrying out an ethnographic study, the sociologists were seeking the complexities and subtleties of the system. They argued that the essence of the system was in the integration of activities but our questions required them to make judgements about isolated activities rather than the overall system. The system should not be considered as a set of loosely interacting individual activities but as a coherent whole which was more than the sum of its parts.

In principle, of course, this viewpoint is undoubtedly correct. In practice, computer systems can't be built without structure and pragmatic decisions have to be made what to support and what to leave out. Effective systems automation requires such judgements to be made. Many current usability problems result from the fact that such judgements have been made with inadequate information by people who have not properly understood the work or its context. The key to building better systems is not to avoid making judgements but to bring in appropriate expertise so that decisions are better informed. Our sociologist collaborators have now recognised that we can only build imperfect systems and that they cannot opt out of making judgements about the process which they are studying.

As well as participating in the design process, we anticipate that sociologists will take on other roles in the systems development process.

1. During evaluation of the interactive system, the sociologists will be involved in studies of the system in use and will be able to assess the impact of the system on working practice.
2. When a system is available for experiment, the sociologist should identify specific training and documentation requirements which will allow users to investigate the facilities of the prototype.

2.2 System Assessment

System assessment plays a vital role in the prototyping approach adopted within our model of development. Initial results of our observational study has already highlighted the reluctance of the majority of users to tolerate a technically unsound system. As a result we feel it is important to carefully plan how and when users are exposed to the system.

As our interface to the user community, the sociologist plays a crucial role in introducing the prototype system to prospective users. A key requirement of an ethnographer is to establish a rapport with the organisational group being studied. Our collaborators have already identified a number of supportive users to act as initial system assessors. These air traffic controllers accept the need for some system automation (unlike many of their colleagues) and will not approach the imperfect prototype system with an inherently hostile attitude.

Once a sufficiently stable system has been developed in line with our initial users a larger scale assessment will take place. In this phase of assessment, sociologists will observe system usage and assess the impact of the system and how the system can develop. Our planned assessment will be qualitative and we will concentrate on gathering overall user impressions and comments about the prototype system. This form of qualitative system assessment within a working context is essential in social systems and we feel it complements existing forms of user interface evaluation.

2.3 System Installation

The development of a computer system represents only one part of the overall design of workplace systems. In addition to insights into the development of the computer system, observational studies conducted by sociologists highlight the important role of work practices and the importance of reconciling the developed system with existing work practice. This requires significant social input when systems are installed in a work setting.

Existing systems of all kinds must accommodate many different ways of working and it is clear that every user idiosyncrasy cannot be supported in an automated system. The understanding of the system derived by the sociologist should be invaluable in highlighting particular training requirements. Training recommendations can relate unsupported tasks to the system and demonstrate how existing work practices can be achieved within the system.

As far as we are aware, sociologists have not been previously involved in studies of training requirements for interactive systems. The implications of asking them to identify appropriate training from the ethnographic work are still unclear and, indeed, this may open up a new research area in sociology.

3. Collaborating Effectively

One approach to collaborative working is for individuals to become experts or, at the very least, to acquire some expertise in both disciplines involved. This approach is favoured by Gilbert (1991) who suggests that the HCI designer must be a computer scientist, sociologist, graphic artist and psychologist. We consider this to be arrant nonsense. Apart from the very high cost of acquiring all these skills, it is by no means certain that the skills needed by a sociologist (say) are the same as those of a computer scientist. Of course, awareness of other disciplines is necessary but we reject the notion that multi-faceted expertise is essential.

Rather, we believe that effective collaboration requires multi-disciplinary design teams who respect the skills of each discipline and who are prepared to make compromises in order to

work together and to enhance the systems design process. In our own case, we believe that we work reasonably well with our sociologist collaborators. Our success is due to a number of factors which we think are critical for effective research collaboration:

1. Equality of objectives
2. Flexibility
3. Mutual respect
4. Informal and formal contacts

We believe that the most important pre-condition for effective research collaboration across disciplines is that the research is of value from the perspective of both disciplines. Research projects which are dominated by one type of research are liable to be unsuccessful as it is inevitable in such cases that the partner with the lesser contribution is seen as providing some kind of service to the other partner. The effect of the collaboration should be to add value to both research areas; it should still be possible to publish in the journals and conferences of both disciplines.

It is a truism that collaboration never works successfully if some of the partners are unwilling to move from some entrenched position. We assume of course that sociologists don't have strong views about computer science and vice versa. However, we believe that it is equally important that each collaborator is not dogmatic about their own discipline. Working together across disciplines often forces you to take a new perspective on your own work. If this challenges a strongly-held opinion, it can be very disturbing indeed. The natural human tendency in such a situation is to blame the collaborators rather than to change the opinion.

Each party in the collaboration must respect the other's expertise and must let them get on with their work without interference. It probably helps if the participants are recognised by their own discipline. The other parties then can be reasonably confident that the people they are dealing with are competent. Worries that the 'sociology'/'computer science' is of poor quality can impair a relationship.

Finally, we consider that both formal and informal contact is necessary to build effective working relationships. Because of the lack of shared background, formal progress meetings are not enough; they need to be supplemented by informal discussions where the participants simply talk about each other's problems, share experiences (problems with computers are universal!) and generally discover common interests.

However, scheduled and formal progress meetings are also essential where the project is reviewed against a plan and the objectives of the following period are established. Without such shared progress meetings, there is a tendency for each discipline to become involved with its own problems and to diverge from the overall objectives of the project.

4. Problems

Our particular collaboration started with a proposal for a research project and it was here that the first obvious problem emerged. We agreed that the most effective approach was for

both groups to present their views of the project then to integrate these to form a coherent proposal. We then found that we didn't have a common model of what a research proposal ought to be.

The computer scientists wrote a proposal with a list of objectives, a method of research and a timetable for that research; the sociologists wrote a proposal consisting of a discussion of the general problem being tackled then a list of specific questions which were of interest in that context. The sociologists deliberately avoided tying themselves down to specific objectives. They made the valid point that the study proposed was a complex one and it could not be decided in advance what the most important objectives might be.

The problem in our case (and, we suspect, in most cases) was easily resolved. The agency sponsoring the research expected a proposal for a science rather than a social science project so we followed the accepted model presenting a list of objectives, research methods, etc. Of course, this meant our social science colleagues had the problem of deciding if we had correctly translated their research objectives into what was to them an unfamiliar form.

During the writing of the proposal and subsequently as we have been working together we have come across a number of other difficulties. Some of these have already been introduced. The major problems we have discovered can be classified under three headings:

1. *Communication* we had to learn something about each other's language.
2. *Methodology* we had to learn about each other's ways of working.
3. *Comprehension* we had to understand the principles underlying each other's discipline.

4.1 Communication

Problems of communication are normal in any inter-disciplinary collaborations as each discipline has its own specialised vocabulary. We believe the problems are greater in cases where a science or engineering discipline is collaborating with a social science. A fundamental problem was that each discipline used normal English words as jargon terms. Practitioners in each discipline had no difficulties in deciding, from the context, when the term usage was in its specialized meaning or when it was in its 'generally accepted' meaning. However, realizing that there was a different specialized meaning was a problem for both disciplines. Some examples of words which caused misunderstandings were 'semantics', 'abstraction' and 'model'.

As an illustration of this problem, we produced a very simple mathematical specification of some abstract data types and stated that this defined the 'semantics' of these entities. In essence, what we were saying was that an abstraction of the meaning of the entities which we needed for this project was set out by the mathematics. Our collaborating sociologists found the notion of a mathematical specification of semantics to be very alien. They argued that, since meanings were socially negotiated it was impossible to capture their sense by mapping them onto a mathematical system.

When talking of 'semantics', we meant the definition of the behaviour of a single entity; the mathematical specification allowed us to communicate this unambiguously to other

developers. The sociologists view was that the 'semantics' of an entity was not simply dependent on the entity but also on the observer of the entity and the context of observation. After a great deal of mutual incomprehension, it became clear that we were each using the term 'semantics' in a completely different and discipline-specific way and we eventually came to see each other's point of view.

4.2 Methodology

The research methodologies in computer science and in sociology are completely different. Computer science has an engineering-oriented approach to research. Its emphasis is on demonstrating the feasibility of concepts by building systems. Abstraction is a key part of this process - the computer science researcher is always examining the problem at hand to discover abstractions and to produce general rather than specific solutions. Confirmation of theoretical predictions or, indeed, the discovery of theories by analysis of experimental results is not currently a significant research methodology.

The sociology research methods of relevance to our work are based on observation. The sociologist looks at a society to determine its practices, and behaviours. The sociologist may try to fit these observations into an existing theoretical framework or may derive a new theory from them. For ethnographers, all detail is potentially significant and they are reluctant to make abstractions in case this hides potentially important system details.

However, different sociologists are concerned with looking for different things. Some sociologists are concerned with observing social behaviour; others seek the relationships between observed features of a society and aspects of social behaviour. Some sociologists are convinced that as well as observing social behaviour they should postulate reasons for it; by contrast, others are vehemently against trying to explain observations and see the role of the sociologist simply as a disinterested observer.

Sociologists are not usually concerned with discovering improved ways of carrying out a task, with devising techniques and methods for supporting a particular set of behaviours or with inventing new ways of tackling a particular problem. They work very hard to avoid being classed as social engineers who pose 'solutions' to observed social problems.

Accepting that the different methodologies of each discipline are equally valid and accepting the need to reconcile them is extremely important. We have reached this stage of acceptance but have not yet found an effective way of merging our ways of working so that the sociological research can be used in a *systematic* way in the system development process.

4.3 Comprehension

Both disciplines have problems in understanding what the other discipline is about (that is, what do sociologists/computer scientists actually do). From the standpoint of a computer scientist we found this a particular problem as we were used to a hierarchic model of knowledge. When learning a new subject, our normal approach is to tackle the problem by starting with elementary texts then reading progressively more advanced material.

Applying this approach to the understanding of sociology was not successful because knowledge in that discipline is not hierarchically structured. Rather there are many diverse and distinct areas such as the sociology of work, the sociology of religion, etc. which, on the surface at least, appear to have little in common. Furthermore, there are several schools of thought in many of these areas based on different theoretical frameworks. We found it impossible to reconcile these different areas and 'schools of thought' in sociology to build a general model of the discipline.

Furthermore, there is no single notion of what sociologists 'do'. Computer scientists have different specialities but have the shared objective of (somehow) building better computer systems. There does not appear to us to be a comparable broad objective which links practitioners in sociology.

Sociologists undoubtedly have comparable problems in understanding the systems development process. A particular difficulty (perhaps resulting from the poor usability of many personal computer systems) is that the sociologists have no conception of how long a particular system change might take. What appears an immensely complex task to them (e.g. changing the colours on a display) can be accomplished in minutes yet apparently simple system changes such as adding a new operation to a system may take several days or even weeks to implement.

5. Conclusions

The fact that sociologists have a role to play in the design of complex computer systems is becoming more widely accepted although we believe that it will be many years before the participation of a sociologist in a requirements team is normal practice. Our experiences, so far, have been positive and they have suggested that it is possible for systems designers to collaborate with sociologists without undue pain and without compromising the aims and principles of either discipline. The ethnographic studies have already revealed a number of subtle system requirements which are unlikely to have been derived from a conventional requirements analysis process.

For example, in studies of air-traffic control, we have discovered that apparently repetitive tasks such as rearranging paper strips representing aircraft being controlled, are a key part of the activity in that the manual manipulation serves to bring problems and details to the controllers attention. Furthermore, we have discovered much cooperation between controllers is implicit and relies on 'at-a-glance' understanding of other controllers' workspace. User interface tailoring which is usually suggested as a 'good thing' would be positively dangerous in this instance.

Computer systems must always be concerned with abstractions rather than minutiae. Computer systems of the future will have to be tailored to working practices but it will always be the case that work will have to adapt, to some extent, to the automated system. To perform a useful role, sociologists must be willing to make judgements as to essential work characteristics which must be preserved in a computer system and those characteristics which may be changed.

The dangers of collaboration with sociologists is that the use of ethnographic studies of work practices will be seen as the 'answer' and the advantages will be over-stated. Engineering and economic considerations will always be as important as tailoring a system to its social context. Sociologists have an important role to play in the design process but, unlike some authors, we do not believe that their role will necessarily be a central one. Rather, we hope they will be accepted as members of a development team whose status will be comparable with that of cognitive scientists and software engineers.

Acknowledgements

Thanks are due to our collaborators John Hughes, Dan Shapiro and Dave Randall from the Sociology Department at Lancaster University. We look forward to their views on us! Thanks also to the UK Research Councils Cognitive Science/HCI Initiative for funding our work and to Thomas Malone for inspiring the title of this paper.

References

1. Bannon, L. and Schmidt, K., (1991), "CSCW: Four Characters in Search of a Context", in J.M. Bowers and S.D. Benford (eds): *Studies in Computer Supported Cooperative Work. Theory, Practice and Design*, North-Holland, Amsterdam.

2. Bentley, R., Sawyer, P., Rodden, T. and Sommerville, I. (1992), "A Prototyping Environment for Dynamic Data Visualisation", Paper submitted to *5th IFIP Working Conference on User Interfaces*, Ellivouri, Finland, 10th-14th August, 1992.

3. Gerson, E. M. and Star, S. L., (1986), "Analyzing due process in the workplace", *ACM Transactions on Office Information Systems*, 4 (3), 257-270.

4. Gilbert, G.N., (1991), "Social and Computer Interaction", presented at the UK Joint Council Initiative HCI Summer School, London, September, 1991.

5. Grudin, J., (1990), "The computer reaches out: the historical continuity of interface design", in *Proceeding of CHI'90 Human factors in computer systems* (Seattle, Washington, (April 1 -April 5), ACM, New York.

6. Harper, R., Hughes, J. and Shapiro, D., (1991), "Harmonious Working and CSCW: Computer Technology and Air Traffic Control", in J.M. Bowers and S.D. Benford (eds): *Studies in Computer Supported Cooperative Work. Theory, Practice and Design*, North-Holland, Amsterdam, 1991.

7. Heath, C. and Luff, P., (1991), "Collaborative Activity and Technological Design: Task coordination in the London Underground Control Rooms", in *Proceedings of ECSCW'91* (Amsterdam, Sept 25- Sept 27), Kluwer.

8. Kyng, M. (1988), "Designing for a Dollar a Day", *CSCW '88. Proceedings of the Conference on Computer-Supported Cooperative Work,* September 26-28, 1988, Portland, Oregon. ACM, New York, N.Y., 1988, 178-188.

9. Norman, D. and Draper, S. (eds), (1986), *User Centered System Design*, Lawrence Erlbaum Associates, New Jersey.

10. Suchman, L., (1983), "Office procedures as practical action", *ACM Transactions on Office Information Systems*. 1, 1983, 320-328.