Socio Technical Systems Engineering Handbook

Contents

Introduction

The problems that arise in procuring, developing and operating complex IT systems are not just technical, engineering problems. These systems are developed and are operated by people, working in organisations. These people and organisations inevitably have different, often conflicting, views on what the system should do and how it should inter-operate with other systems. Different organisations have their own ways of working, goals and culture. The IT system is therefore part of a broader âĂŸsocio-technicalâĂŹ system and many people are increasingly convinced that we have to think about systems engineering from this broader socio-technical perspective if we are to improve the quality and effectiveness of our IT systems.

Socio-technical systems engineering (STSE) is a set of methods and techniques that support the technical processes of systems engineering. These methods and techniques help systems engineers understand the socio-technical issues that affect the systems being designed and operated and provide help in taking these socio-technical issues into account when making procurement, specification and design decisions. Like all systems engineering, STSE relies on judgement and creativity and cannot be simply represented as a standard set of processes or best practices.

This Handbook

The aim of this handbook is to summarise almost 20 years of research and development in socio-technical systems engineering. We have designed the handbook to be accessible to both practitioners and researchers. The handbook is organised as a set of short chapters that explain the issues in the chapter to practitioners. Each chapter is backed up by an appendix, which includes summaries of and links to relevant research in the area. These appendices can be found online on the **LSCITS STSE Handbook website** at `http://archive.cs.st-andrews.ac.uk/STSE-Handbook/`.

We have also collected together a number of related websites, originally hosted at Lancaster University which provide supplementary information to the material in this Handbook. These include the Patterns of Interaction website and a tutorial on carrying out fieldwork. We also maintain local copies of as many related papers as possible on the website so that there is no reliance on these continuing to be available elsewhere.

Over the past 20 years, many colleagues and friends have contributed to this research. We have listed as many as we can think of in the acknowledgements pages on the website - apologies to anyone who has been left out.

Background

This handbook presents socio-technical systems engineering from a particular perspective that was originally developed at Lancaster University in England in the early 1990s. The software engineering research group, led by Ian Sommerville, was interested in requirements engineering research and this group got together with people in the Sociology department, led by John Hughes, who were interested in the sociology of work and who were increasingly interested in how computer systems were really used in the workplace.

Our interests at that time were focused on how ethnographic studies of work practice could provide information that informed the design of systems to support work. We were involved in ground-breaking research in this area where we studied the practice of air traffic control. Over time, our interests diversified into a wide range of application domains (control rooms, banking, healthcare, etc.) and we looked at how ethnographic techniques could be adapted to be used alongside systems engineering processes. At the same time, ubiquitous computing was emerging as an important research area and the Lancaster work was unique in that it took a socio-technical view of ubicom rather than the more prevalent device-oriented perspective of most researchers in this area.

All of this led to involvement in two important projects in the early years of the 21st century. These were so-called Interdisciplinary Research Collaborations involving researchers from different disciplines and universities. The **DIRC project** (`http://www.dirc.org.uk`) focused on socio-technical issues affecting the dependability of complex software systems and the **EQUATOR project** (`http://www.equator.ac.uk`) was concerned with ubiquitous computing. Since then, work on socio-technical systems engineering has been carried on in the **LSCITS project** (`http://lscits.bris.ac.uk`), which has sponsored the development of this handbook.

The key distinction between the work at Lancaster and other work in this area by researchers in social informatics, CSCW and HCI was that our work has always had the practical goal of (ultimately) influencing the way that complex systems are engineered. Obviously, we have been influenced by other work such as that on soft systems methods, participative design, cognitive systems engineering and social informatics and we certainly donâĂŹt claim that we were the originators of all of the ideas discussed here. We believe that these alternative perspectives, which we summarise in this paper have much in common with each other and with what we do but we do not have the time or effort to include all of it in this handbook.

The Lancaster team has now largely dispersed and the focus of work on socio-technical systems engineering has now moved elsewhere. Some people have retired, others have moved on to do different things but all of us will remember our time at Lancaster with affection and how we did much more than play lip service to the notion of inter-disciplinary research.

Chapter 1

Socio-technical Systems

Gordon Baxter, University of St Andrews

Summary

Many people now acknowledge that systems which are developed using a socio-technical approach are more likely to be acceptable to end users and to deliver real value to stakeholders. Socio-technical approaches can help the design of organisational structures and business processes as well as technical systems. Even though most systems can now be described as socio-technical systems (STSs), socio-technical approaches to development are not widely used. Most developers still follow the traditional, reductionist approach to development, which divides the system into a social system and a technical system. Such an approach fails to consider the ways that the social and technical aspects are interdependent and interact, which is central to the performance and behaviour of STSs. This chapter discusses the development of socio-technical systems design and discusses some of the problems that have hindered the use of socio-technical approaches.

Background

The concept of socio-technical systems emerged shortly after World War II, in work being carried out by what is now the Tavistock Institute. They were looking at long wall coal mining operations, and had identified that the way that people worked did not follow the mechanistic view of work, which emphasised specialisation and the division of labour. Instead they found that the social aspects were also important, particularly the ways that individuals and teams co-operated and collaborated to use the available technologies. The performance of the system was based on the ways that people worked with machinery in that particular context.

The ideas of STSs were taken up in many countries across the world, and several philosophies emerged, which generally reflected local and national cultures. In Scandinavia, for example, they emphasised the humanistic aspects, which reflected the culture of workplace democracy in those countries. These methods remained largely unchanged, though, as new ways of working, and new types of organisational structures emerged. Socio-technical methods were largely sidelined when lean production techniques and business process re-engineering emerged in the late 1980s. The ideas behind the methods, however, have remained relevant, and there have been some attempts at bringing the methods up to date by linking them to agile approaches to system development, for example.

ETHICS

One of the best known socio-technical methods is ETHICS (Effective Technical and Human Implementation of Computer-based Systems). ETHICS was developed in the UK in the early 1980s by Enid Mumford from the Manchester Business School. Like most STSD methods, ETHICS considers the introduction of a new system as part of a broader change process. There are four identifiable aspects to this change process that need to be considered:

1. Setting and achieving system objectives that take into consideration the differing views of all of the system stakeholders.
2. Adaptation to the new system and new ways of working.
3. Integration of the various elements of the new system into a coherent, functioning whole.
4. Stabilisation of the new ways of working.

ETHICS implements the change process using a series of steps. These deal with organisational issues, as well as providing some guidance for the design of the final system, including the design, implementation and evaluation of the technical aspects of the system:

- The diagnosis of the economic and social needs, covering both efficiency and job satisfaction.

- The setting of efficiency and social objectives.
- The developing of alternative strategies that meet both sets of defined objectives, and selection of the strategy that best meets both sets of objectives.
- The detailed design of the chosen strategy.
- The implementation of that design.
- The evaluation of the implementation of the system after it has bedded in.

The identification of the technical optionsâĂŤthe hardware, software and the design of the human-machine interfaceâĂŤis carried out simultaneously with the identification of the organisational options (i.e., the different ways of structuring the organisation to meet the efficiency and job satisfaction objectives). The sets of technical and organisational options are usually interdependent.

What is evident from methods like ETHICS is that the development of any system is necessarily interdisciplinary work. The people and the organisation cannot be considered in the same mechanistic way as the technology, and require different skills and disciplines to analyse and design the social aspects of the system, and the ways in which they interact with the technical aspects of that system.

Other approaches encompassing STS ideas

There are many other methods that are explicitly described as socio-technical methods or approaches, apart from ETHICS. There are also several other approaches that encompass socio-technical ideas, or at the very least are consonant with those ideas. These methods include:

- Soft Systems Methodology.
- Cognitive Work Analysis.
- Ethnographic workplace analysis.
- Cognitive systems engineering.
- Human centred design (video) . For software, this is often called user-centered design.

Problems with existing STSD approaches

Many of the existing STSD approaches reflect their cultural roots in terms of time, space and place, and hence have not always been transferable to other situations. There are several readily identifiable problems with existing approaches, including:

- A lack of consistent terminology, even to the extent of variations in the definition of exactly what constitutes an STS.
- Determining the appropriate levels of abstraction to use, based on where the system boundaries are drawn, and a tendency to focus on the technical aspects in greater detail.
- Conflicting value systems, with humanistic values on the one hand, being regarded by some as incompatible with managerial values.
- A lack of agreed success criteria, partly because it can be difficult to identify evaluation criteria for the social aspects of the system.
- A focus on analysis rather than synthesis, showing how a system looked once it was built, rather than offering support for how to construct a successful system.
- A lack of multidisciplinarity, with some disciplines failing to understand what other disciplines can contribute to system development.
- A perceived anachronism, because the methods did not change to reflect the changing nature of organisations and ways of working.
- A lack of support for identifying the appropriate stakeholders and users.

Implementing Organisational Change

The introduction of a new system often forms part of a larger organisational change process. Sometimes the change will be evolutionary, and sometimes it will be deliberately revolutionary, such as when the company wants to introduce new ways of working that may lead to reductions in the workforce. The group whose main objective is to bring about the organisational change often have extensive backgrounds in business management, and a good understanding of business processes; the systems development group, in contrast, will usually have a strong technical background. Implementing organisational change, however, can often give rise to unanticipated effects that may have an impact on other parts of the business. In some cases this will have a knock-on effect on the development of the system that is to support that business. It is therefore important that the organisational, social and technical aspects of the system are considered together, and that the organisational change team, and the system development team communicate and synchronise their activities on a regular basis. STS methods provide one way of at least ensuring that the organisational, social and technical aspects are appropriately dealt with. We explain how a socio-technical approach can link the processes of system development and organisational change management in our chapter on Socio-technical Systems and Organisational Change.

Retrospective

The ideas that underpin socio-technical approaches have never really gone away, even though some of the methods may have fallen into disuse. In particular, when lean methods and business process re-engineering came into fashion, STSD methods were perceived as being unable to deal with the new world. In spite of this, there has been increasing awareness of the need to consider systems as socio-technical systems, although many people fail to fully comprehend the importance of considering the social and technical issues together. Instead, they take a more traditional reductive approach to system development by dividing the system into a social part and a technical part. This state of affairs can be addressed by sensitising stakeholders to the concerns of other stakeholders, and convincing them about the value of adopting a socio-technical approach, and by integrating STS thinking into the systems development and organisational change management processes.

Chapter 2

Software Development and Deployment

John Rooksby, University of St Andrews

Summary

Systems development is more than a technical procedure; it is a form of cooperative work. The development of any non-trivial system involves various kinds of planning and procedures, necessitates forms of distributed coordination work and requires some subtleties amongst workers in the form of awareness of the work of others. These practices are intricate and fine-grained and saturate every level of software engineering from coding, to testing, to documenting, to procurement and marketing.

The related chapter on Requirements and Design focuses specifically on these process activities and how socio-technical analyses can contribute to them.

Background

Software and systems development work has evolved and transformed over the decades. Many of these changes relate to the need to construct, manage and assure increasingly large and complex systems. The changes include:

- A shift in focus from programs to systems, and from technical to socio-technical problems;

- A shift from development to procurement, configuration, and reuse, and from âĂIJgreenfieldâĂİ engineering to âĂIJbrownfieldâĂİ engineering where systems must be integrated and coordinated with other systems;
- A shift from small to large development teams that are spread between sites and often between different organisations, and an increasing recognition that the communication and cooperation between developers needs to be intensive and of quality;
- Changing user-provider relations, with iterative development and reduced time to market meaning issues are often knowingly left until post-deployment, and usefulness becoming relevant alongside or in place of correctness;
- Development increasingly becoming a professional activity made accountable to organizations and regulatory bodies;
- Technology transfer from research entailing the reorganisation of current practices and the acquisition of new skills.

Our work has been driven by two concerns:

1. Research in software and systems engineering has mostly focused on technology and has paid little regard to the fact that human, social and organisational issues have a major influence on all aspects of engineering processes.
2. The focus of the software and systems engineering research community has mostly been on âĂŸinterestingâĂŹ systems such as complex, safety-critical systems with requirements for advanced methods of development and validations. However, most systems are more mundane (although they may be complex in different ways) and methods of developing such systems (e.g. the extensive use of ERP systems) have changed radically over the past 20 years. These changes have not been widely acknowledged by the research community.

Our concern has been that while the work of systems development has changed, the academic discipline of software engineering has remained static and can often miss real world problems. There is a gulf between academia and practice, caused not just by the failure of organisations to heed the lessons and insights from software engineering research, but also problems with the relevance of this research to real-world practice.

Organisational Issues

Many of the problems of software development are organisational; they are the problems of coordination, scheduling, decision-making, awareness, and so on. A key difference between systems engineering and socio-technical systems engineering is that the latter takes these into account.

Systems development is normally managed on a project basis. Projects are formatted organisational arrangements within which people and resources can be allocated, coordination tools and procedures deployed, and they provide the context for the organisational accountability of system development, particularly the measurement of progress. Contrivances associated with projects can include phases, specifications and plans. Other contrivances can include allocating roles and organising people into teams, specifying means of cooperation such as regular meetings, sign-off and so on.

Contriving the orderliness of work does not in itself ensure this orderliness or provide remedies for all contingencies. For example, plans are followed dynamically and remade as development progresses. Questions repeatedly arise during the development and testing of systems as to what exactly can be done to satisfy the plan, what parts of the plan are achievable given the time available, and what is missing from the plan.

Cooperative work in systems development and testing is often kept orderly through the use of âĂIJordering devicesâĂİ. These ordering devices may be information technologies such as versioning systems, wikis, and workflow management systems. They may be paper-based technologies, such as task-cards or sign-off sheets; or, they may be more procedural such as verbalising particular actions or events, working in rotating pairs, holding meetings, or the adoption of a particular coding style.

Managing the ordering devices in software engineering is often perceived as bureaucratic work that does not contribute directly to the system itself. However, this work is always crucial to the successful development of any non-trivial system. It is therefore important to notice and analyse the repertoire of ordering devices used in any particular systems engineering project, the interdependencies between these, and their strengths and fallibilities.

There is also a relationship between organizational structure, organisational priorities and the ways in which systems projects are carried out. This stretches well beyond what kind of method an organisation selects (agile, plan-based, etc.) and into how any particular method is practiced. The ways in which an organisation is structured can significantly impact on the way a project is practiced and, indeed, the architecture of the system itself. This can be in terms of hierarchy and decision-making. It can be in the ways in which collaboration and teamwork are structured. And, it can be around the availability of people and resources.

The priorities of an organisation also significantly affect the ways in which systems engineering is practiced. These often become apparent as deadlines approach, with the need to make a profit, or make software available for pre-scheduled activities overtaking the concerns for producing reliable or fully functional software.

Specifications

Systems specifications take different forms depending on the design method used and whether the project is being done in-house or by an external supplier. In whatever form they take however, specifications provide a framework within which, and in reference to which, design and testing, and user-designer relations, get worked out in practice.

- **Contracts**: At one extreme, specifications can take the form of a bulky contract between a supplier and customer stipulating the work that will be done over a period of years. Formal contracts between a supplier and an organisation, the formal, legal stipulation of work and responsibilities, are more than simply statements of fact but get dragged into everyday work, invoked, pointed to, metaphorically waved about, and used in a number of ways. The contract is a living document, a constant source of reference and discussion around which work and activities get organised, changed, modified and abandoned.

- **Cards**: At the other extreme, agile methods produce short-term requirements written on small cards throughout a project. The cards used in agile programming may be less formal than a contract, but these too are more than statements of fact; they are a source of reference and discussion serving both to structure and coordinate a project team. These cards are not only used within a process, but embody that process, supplying physical devices that can be repeatedly arranged around in the ongoing coordination of work activities.

As with any kind of plan, the development work and the system actually produced differ from what is stipulated in the specification. The actual project work and the finished system are instead a product of putting the specification into practice. This involves working out how the specification translates into, and relates to, the multifarious activities of development work, and the specifics of the emerging system. These activities, decisions and appraisals are often fashioned through intense negotiation between the different parties, in contingent and rapidly changing circumstances, in which the specification is a key feature and resource.

The area of specification is one where socio-technical factors are, perhaps, most evident and so it has been a focus for research in using socio-technical analyses. The chapter on Requirements and Design covers some of this research and its applications.

User-centred design

Systems development should orient to how the system will be used, what functionality is needed, what infrastructure and resources for running the system will be available, and what the usability issues are. Requirements engineering, particularly in user-centred design methods, often seeks to improve the quality of user-relevant information available during the design process.

This is important, but our experience is that user-centred design methods are too idealised. The realities are:

- **User participation**: The reality tends to be that where users are involved, these are often the âĂŸexpendableâĂŹ people within an organisation (i.e. the ones with enough time to participate) and they find it very difficult to articulate what it is they want from a system. Participatory design is often also abandoned as deadlines start to bite. Users are often also involved in testing systems, but this seems to get conflated with training, which can mean neither is done properly.

- **Customer participation**: The user and the customer are rarely the same. While the rhetoric is systems engineering is often about user centeredness, the reality is that systems engineers must prioritise satisfying the customer. The customerâĂŹs priorities can often be more associated with cost and deadlines than with usability.

- **User and customer proxies**: In many cases, the user or customer is not actually available and so will be simulated. This may be through the creation of user models, but is more commonly done through someone acting as a proxy. In particular, product companies do not always have a pre-existing customer base (and even if they do, need to focus on the expanding the market to other customers) and so some member of the development or marketing team will usually stand in for the customer.

- **Typification**: Whether users participate or not, a substantive part of a systems project involves speculating and reasoning about what users might do with the system. Where no genuine user is available, this will involve talking about what users may do. Where users are available, that user still needs to reason about what they might do with the system and how representative they are of other users.

Pervading the user-designer relation in systems development are issues of generalisation. How does one personâĂŹs needs and opinions generalise to others? How do the issues in one organisation generalise to the issues in others (as potential customers)? How can a product developed for one niche be generalised for a wider market? Systems engineers, even if they have âĂIJusersâĂİ to hand, will inevitably have to engage in some practical social reasoning about how to satisfice the needs of users.

Software testing

Testing, as with every other aspect of systems development, is saturated with social and organisational issues. We have found that while developers seem comfortable acknowledging the social and organisational issues in, say, requirements engineering, they are still extremely reluctant to acknowledge that similar factors pervade testing.

We have undertaken studies to characterise testing as it is done âĂŸin the wildâĂŹ. We have not focused on newsworthy achievements or experiences in testing and have purposefully not discussed safety critical testing. We are certainly making no claims that the examples represent best practice. What we have achieved is a characterisation of run-of-the-mill testing, one that can supply insights into the kind of work that is currently done in many organisations on a mundane basis. This kind of testing is not usually safety critical but is often project or business critical.

We have identified a number of themes:

- Plans are followed dynamically and remade as testing progresses. This is because testers work with limited resources, but also because problems are routinely discovered during testing that can demand reformulation of the plans.

- Testing involves work to stay organised, with coordination of effort between testers and between testers and the broader development team being a demanding concern.

- Time is of constant relevance and a significant factor in the way testing is organised. Decisions on whether the time and effort are justified are essentially and contingently organisational.
- There is congruence between organisational structure, organisational priorities and the way tests are performed. For example the people and locations available for testing, and the priorities given to release dates, particular customers and so on heavily shape testing.
- Tests are attributed significances. Not all possible tests are undertaken, we have seen testers choose which are the most significant to do given the time available.
- Testing involves reasoning and speculation about practices and situations of use. A substantive part of a systems project involves reasoning about what users might do with the system. Therefore the practical sociological reasoning of testers is not limited to how to coordinate during the course of testing, but is central to deciding what it is a reasonable test to set.

In the face of real world complexity, testing is a satisficing activity. Systems validation and verification can never ensure the correctness of a real world system. Systems engineers have to find and accept âĂŸgood enoughâĂŹ solutions, not because less is preferred to more but because there is no choice.

Retrospective

In our studies, we have tried to achieve a characterisation of run-of-the-mill development, one that can supply insights into the kind of work that is currently done in many organisations on a mundane basis. We believe that a better understanding of the everyday work of systems development helps us understand why technologies are and are not used and can inform the design of more usable methods and technologies.

We have deliberately steered away from safety critical systems, focusing on ones that are transformative, ones that are often project or business critical. These are the kinds of systems that are overwhelmingly common, and for which best practices can be distorted by the preoccupation in software engineering with the safety critical. It is common in the literature to use stories of good and bad practice, stories of the kind of work systems engineers should aspire to or avoid at all costs. We have trodden a different path, using examples of the kinds of work that we believe will be recognisable to anyone with experience in real world systems development.

Chapter 3

Requirements and Design

Ian Sommerville, University of St Andrews

Summary

Much of our work in social analysis of complex systems has been concerned with how we can use such analyses in systems specification and design. In this chapter, I discuss how fieldwork can be used to gather date that informs the requirements engineering process. I discuss the ways in which fieldwork can be used in conjunction with system prototyping and how analyses can be used for the âĂŸsanity checkingâĂŹ of requirements. I conclude by discussing the limitations of fieldwork in informing complex system requirements but suggest, nevertheless, that short observational studies should be an inherent part of requirements engineering processes.

The chapter on Coherence discusses a method to support the use of fieldwork in requirements engineering and the chapter on Patterns captures common features of work settings that can be used to help understand system requirements.

Background

Requirements engineering (RE) was one of the starting points for our work in socio-technical systems and interactions with our colleagues from Sociology. We were interested in the general problems of understanding the requirements for complex IT systems and in developing new viewpoint-oriented approaches to RE. But it became clear to us that many of the practical problems that people encountered with systems were a consequence of the ways in which they worked and that if we were to understand their âĂŸrealâĂŹ requirements, then we really needed a better understanding of work, the actual rather than the formal business processes and the relationships between these processes and organisational factors that influenced the ways in which work was done. Consequently, we started to explore how ethnographic approaches could be used to understand work and to investigate how fieldwork could be used to inform the requirements engineering process. If we understood work as it really was done, then we were convinced that we would produce higher quality software system requirements.

Of course, we know of some of the views of the agile community who suggest that requirements should emerge incrementally during development. While this is perhaps true for some kind of requirements, the reality of current systems engineering is that some kind of requirements document is always needed for large systems before development begins (and sometimes before the development contract is issued).

Understanding requirements

There are two problems in requirements engineering where ethnography may help:

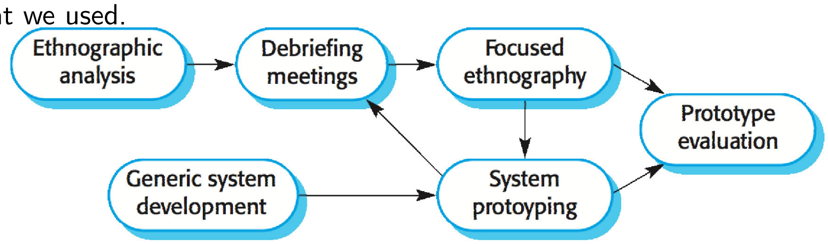
1. **Businesses do not understand their own processes**. Requirements for software are often based on an understanding of a âĂŸformalâĂŹ process which bears little relationship to the real business processes that are used. Ethnography can reveal the differences between formal and actual processes and so allow more appropriate requirements to be proposed.

2. **People cannot articulate their requirements**. Ethnography can provide information about what people actually do and so can serve as a source of requirements. This is particularly useful when combined with a prototyping approach as discussed below.

However, fieldwork simply develops a rich picture of how people work and the business processes involved. It does not, in itself, generate system requirements so we need to have one or more mechanisms to help translate the understanding of work into practical requirements for system support.

Fieldwork and prototyping

Prototyping is an established technique for supporting the requirements engineering process. A system prototype is developed and is used as a basis for experiment. People find it much easier to articulate what they need when they have a real system in front of them. However, a major problem with prototyping is getting user involvement and so we explored how fieldwork can be used to inform prototype development. The following diagram illustrates the process that we used.



We identified 4 key questions that we should ask fieldworkers who have been engaged in studies of work:

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

2. What are important **current activities** which need not be supported in an automated system because the activities are a consequence of the fact that no or limited automated support is available?

3. What **characteristics** of the existing system must be **replicated** without change in a new system?

4. What **activities** from the existing system may be **supported** in a way which is different from that used in the current system?

We developed these questions during our initial studies of air traffic control but we think that they are still the key issues in translating an understanding of work to system requirements.

Sanity checking

The reality of systems development is that requirements are not necessarily going to be informed by fieldwork so they may be based on an inadequate understanding of way in which people really do their work. An approach to requirements engineering based on social analysis can be helpful here as it can highlight pitfalls and things that should not be in the requirements.

This can be a cost effective way to use fieldwork in the requirements engineering process as the requirements focus the fieldwork âĂŞ rather than building a general picture of the work being done, the fieldworker can focus on the activities that are reflected in the requirements and can identify requirements which could cause problems in practice.

A situation where we used this approach was in a financial institution that wanted to introduce a new counter system. We discovered that the requirements were such that they required the teller to enter all of the customer information without interruption âĂŞ something that is completely impractical in a busy high-street branch. After our studies, the requirements were changed to allow information to be added incrementally.

Viewpoints

We have always adopted the position that, while the ideal fieldworker is a trained ethnographer, we will only be able to introduce fieldwork into requirements engineering processes when we provide support for non-specialists to do this work. The Coherence method and the work on Patterns of Interaction all reflect this view.

Our starting point for developing these approaches was earlier work done on âĂŸquick and dirtyâĂŹ ethnography where we departed from the conventional notion that ethnography should be a prolonged process and developed an approach to ethnography that was intended to generate useful information quickly. This could be requirements sanity checking as discussed above or could be a short period of fieldwork that focused on understanding the system in general and the types of problems and issues that arose. We discovered that even a short interaction was very valuable in illustrating the complexity of some processes.

To help with this short period of fieldwork, we came up with the notion of social viewpoints as a way of organising the fieldwork and its documentation. These social viewpoints are, essentially, perspectives on a system and we recommend three such viewpoints:

1. The **work setting**, which describes the environment where the work takes place and the interactions with this environment. This is often reflected in the way that work is physically organised to allow, e.g. to facilitate communications between the people involved in the processes. It may also take into account the use of shared electronic resources such as shared folders on a server.

2. The **work flow**, which describes the sequences of work activities, information flows etc. The important thing here is to look at how the flow of work is used to coordinate the work of the people involved and to look for how people handle exceptions that arise.

3. **Social and organisational perspectives**, which show how the work of individuals in the process relates to other peopleâĂŹs work and to broader organisational issues. Therefore, you might look at the effects of the need to comply with regulations on how the work is done, how people become aware of what other people are doing, etc.

Documentation

Fieldwork is usually recorded as a set of notes in narrative that is perhaps supplemented by photographs of the work setting, documents about the work, video recordings, etc. This body of work is quite personal to the fieldworker so it needs them to be available to interpret it. This is obviously problematic as they cannot always be available for consultation so we looked at alternative means of documenting the work.

We developed a tool called the DesignerâĂŹs Notepad which was, essentially, a simple tool that allowed a user to cut and paste information from the fieldwork record into multimedia notes and to link these notes together. These could be set up by the fieldworker and then consulted by others involved in the RE process.

This was a one-off tool and is no longer supported but you can replicate much of its functionality by using mind mapping or brainstorming software.

Retrospective

An accepted problem with fieldwork is that it tells you about work as it is done and doesnâĂŹt really give you any clues about ways of doing things differently. Of course, a more informed picture of the work being done means that, hopefully, you will come up with better system requirements. Consequently, we now think that fieldwork is not an activity that should precede the development of system requirements but should not be started until an outline set of requirements is available. It can then be used for sanity checking as discussed above but also for adding details to high-level requirements.

Much of our early work on ethnography and requirements involved work in control rooms or other settings where everyone worked together in the same place. The reality of modern work is that it is often distributed both in time and place so situated fieldwork is much more difficult. These difficulties are exacerbated by the fact that distributed work is facilitated primarily by electronic rather than paper documents and the ways in which electronic documents are used is harder to observe.

The other problem with fieldwork, which is shared with user centred approaches to requirements such as those used in agile methods, is that it is mostly concerned with work as done by users. It is therefore less useful for understanding broader organisational requirements or what are sometimes called âĂŸnon-functionalâĂŹ requirements âĂŞ dependability, security, compliance, etc.

Nevertheless, in spite of these problems, we are convinced that a short period of fieldwork can be immensely valuable in the requirements engineering process. By observing how the work to be supported by a software system is actually done, you can identify key activities that must be supported and can discover problems with proposed requirements before these are implemented.

Chapter 4

System Dependability

Ian Sommerville, University of St Andrews

Background

Complex software-intensive computer systems now run all aspects of our society and critical infrastructure from businesses to the power grid. Many of these systems have to be continuously available and must operate with few or no failures. Unavailability or system failure can mean that the organisation running the system loses business, incurs additional costs or, in the worst case, people are harmed by the system failure.

The term dependability was proposed to cover the related systems attributes of availability, reliability, safety and security.

1. **Availability**. Informally, the availability of a system is the probability that it will be up and running and able to deliver useful services to users at any given time.

2. **Reliability**. Informally, the reliability of a system is the probability, over a given period of time, that the system will correctly deliver services as expected by the user.

3. **Safety**. Informally, the safety of a system is a judgement of how likely it is that the system will cause damage to people or its environment.

4. **Security**. Informally, the security of a system is a judgment of how likely it is that the system can resist accidental or deliberate intrusions.

These are not independent system characteristics. Systems that are insecure can be attacked and their availability can be compromised; systems that are unavailable may not provide essential safety checks; systems that are unreliable may have to be taken down for repairs and so become unavailable.

Since the 1980s, there has been a large body of work looking at technical aspects of dependability. This has been based on two notions:

1. That system failures result from faults that have been introduced during the development process.

2. That the number of system failures can be reduced by avoiding the introduction of faults in the first place, by detecting faults before the system is put into use and, in some cases, by using run-time mechanisms to tolerate faults if they occur.

The technical developments since the 1980s have meant that the dependability of software has, across the board, increased very significantly. For some classes of critical system, such as control systems, the application of dependable software engineering techniques has meant that we can now build software that functions very reliability with a very high level of availability.

However, for enterprise systems, software systems that are typically used to support many different functions in an enterprise, with different classes of user, our track record on dependability improvement is much poorer. These systems still commonly fail to deliver the expected services to their users.

The fundamental reason for this is that all our technical methods of dependability achievement, based on fault avoidance, detection and tolerance, all rely on their being agreed system specification. This specification has to accurately reflect what the software is really required to do.

For control systems and for systems, such as air traffic management systems which are highly proceduralised, it is possible to write a definitive system specification. Most of the systemâĂŹs services are based on procedures or on reacting according to measurements of the systemâĂŹs environment. However, for enterprise systems, with diverse user groups, it is practically impossible to construct a definitive specification that is relatively stable and meets the needs of all users.

The reason for this is that the needs of users are not necessarily consistent and may conflict. So one group of users (group A) may require another group (group B) to enter information but group B may not wish to enter such information as they do not need to do so for their job. Whatever is specified, either Group A or Group B will be unhappy.

Enterprise software systems are part of broader socio-technical systems and the human, social and organisational issues in these socio-technical system profoundly influences both the use of a software system and the views of users on that system. Many so-called âĂŸfailuresâĂŹ of enterprise systems are not technical failures in the sense that there are faults in the system, but are judgements made by one or more user groups about the effectiveness of the system in supporting their work.

For this reason, we are convinced that there is little point in extending technical approaches to dependability achievement for such systems. The problem does not lie in the way that the software is build but on its fit with the organisation and the work done in that organisation. To improve dependability, we need to take a socio-technical approach where we try to develop a better understanding of the settings in which a software system is used, the services needed by its users, how the system supports work and the organisational goals in introducing and operating the system.

The approaches that we can use to carry out a socio-technical analysis are covered elsewhere in this handbook and here we will simply focus on two important issues in socio-technical dependability:

1. The nature of failure, where we argue that failure is a judgement rather than an absolute.

2. The importance of designing technical systems to allow the broader socio-technical system to recover from what is judged to be a technical system failure.

The nature of failure

The drive in technical approaches to dependability has been to avoid system failure by ensuring that faults are not introduced into a system or are tolerated during operation. There is an assumption that failures can be recognised as the systemâĂŹs behaviour deviates from its specification. From this perspective, an observer can examine a systemâĂŹs behaviour and decide whether or not a failure has occurred.

When we look at systems from a socio-technical perspective, however, failures are not so simple. Whether or not a system has failed cannot be decided objectively but depends on the judgement of the system user who has been exposed to the systemâĂŹs behaviour.

The reasons for this are:

1. Specifications are a gross simplification of reality for complex systems âĂŞ its is practically impossible to specify everything that a complex system should and should not do.
2. Users donâĂŹt read and donâĂŹt care about specifications. They therefore donâĂŹt know what the system is expected to do.
3. Because there are multiple stakeholders in a system, behavior that one stakeholder wants may be unacceptable to some other stakeholder. One sees desirable behaviour, the other sees a failure.

There are several factors that influence a userâĂŹs judgement of whether or not a system has failed. These include the observerâĂŹs expectations, the observerâĂŹs knowledge and experience, the observerâĂŹs role, the observerâĂŹs context or situation and the observerâĂŹs authority.

The socio-technical nature of failure means that it is impossible to build a system that will not fail. Changing contexts of use of a system mean that system behaviour that is acceptable at some point in time, may be deemed a failure because of changes in the way that the system is used. Furthermore, because different people have different expectations of the system, it is practically impossible to satisfy all of these âĂŞ some will, inevitably, be judged unacceptable.

Designing for recovery

It is indubitably the case that the trigger for many system âĂŸfailuresâĂŹ is some human action, which is, in some way, erroneous. The human action triggers a sequence of events in the system that ultimately leads to failure. For this reason, some advocates of technical approaches to dependability suggest that replacing humans with automated systems will lead to dependability improvement.

However, we should also remember that, as well as contributing to errors, system failures are often avoided through human actions or checks. People have a unique characteristic to reason about situations which they have not seen before and to take actions in response to these.

Furthermore, after some system has failed, recovery actions are necessary and these are designed and implemented by people. Hence, maintaining human operators and managers in systems is essential in our view for long-term dependability.

As we have discussed,s system failures are unavoidable so it will become increasingly important to design systems to support recovery. Recovery often involves people taking actions which are, in some way, abnormal. For example, if a particular file is corrupt and causing system failure, deleting the file may be the best thing to do. This may lead to a conflict between recovery and security. Users have to take actions to recover but the security features in a system stop them doing so. Finding the right balance between recovery support and security is a difficult challenge.

Some general design guidelines that we have developed to support recovery include:

1. **Avoid automation hiding**. This means that information should not be hidden in a system and only revealed to users when the system believes it necessary. It also means documenting configurations and making public where information is stored in the system. Users should be able to access information about the system and its state so that they can be informed about recovery decisions.

2. **Provide an emergency mode of operation**. In an emergency mode of operation, normal checks that stop users doing things should be switched off and normally forbidden actions allowed. However, actions should all be logged and it should be made clear to users that they may have to justify steps that they have taken.

3. **Provide local knowledge**. Local knowledge is often incredibly valuable when recovering from problems so systems should include lists of responsibilities, should maintain information about who did what, the provenance of data, etc.

4. **Encourage redundancy and diversity**. For efficiency reasons, maintaining copies of information in a system is often discouraged. However, redundant information is often immensely valuable when recovering from failure. It also makes sense to maintain some of these copies in different forms âĂŞ paper copies, in particular, can be useful as these are accessible even without power.

Chapter 5

Cognitive Systems Engineering

Gordon Baxter, University of St Andrews

Summary

Cognitive systems engineering is an approach to socio-technical systems design that is primarily concerned with the systemâĂŹs behaviour âĂŞ what it does and why? This contrasts with other perspectives on socio-technical systems whose emphasis is on structure and relationships between the system agents. It has been primarily used in domains where systems are safety-critical. In use, it focuses on an analysis of how people cope with complexity, understanding how artefacts are used and understanding how people and artefacts work together to create Joint Cognitive Systems.

Background

Cognitive Systems Engineering (CSE) was proposed in the 1970s but was only fully formulated in the early 1980s by people from the field of human factors and ergonomics, and the cognitive sciences. CSE applies a functional approach to the study and development of human-machine systems, focusing on what the system does, rather than how it does it. When CSE practitioners talk about machines, they mean any artefact that has been designed for a particular purpose. Systems are analysed in terms of their joint cognitive systems in order to emphasise the central idea of the co-agency of the human and the machine, rather than simply focusing on the interaction between the two.

CSE largely grew out of work in industrial control systems, particularly nuclear power and now mostly concerns itself with complex application domains (aviation, health care, nuclear power and so on). Its focus is on how systems cope with complexity, how artefacts are used, and the joint cognitive systems that are involved. The goal of CSE is to identify the things (people, resources, artefacts and so on) that are necessary to support the observed ways in which people and systems work, and to use these insights to inform system development.

CSE was developed to deal with three particular problems that were becoming apparent as computer-based systems came into widespread use in the 1970s:

- The increasing complexity of socio-technical systems, which was largely due to large-scale computerisation.
- The ways that the new technologies were being inadequately deployed, leading to a rise in problems and failures (such as the nuclear accident at Three Mile Island in 1979).
- The limitations in existing models for designing and describing systems. These models were mostly linear, and limited in their applicability, such as the paradigm that viewed people as relatively simple information processors.

One of the aims of CSE was to try to bring about a paradigm shift in the way that people thought about developing interactive systems. The early work on interactive systems had focused on the more physical aspects of the work that needed to be done, such as the studies of socio-technical systems carried out by the Tavistock Institute on long wall coal mining. The increased use of automation and computers, however, had started to shift the emphasis from physical work to cognitive work, which was something that CSE directly addressed. CSE also explicitly recognised the concept of emergent behaviour, noting that systems were more than just the sum of their parts, and how system behaviour is an ongoing process that follows a continuous cycle, rather than made up of single (apparently) unconnected interactions.

CSE considers all systems to be socio-technical systems, so can be considered as another perspective on socio-technical systems design. Rather than focus on structural aspects and the relationships between the agents in the system, CSE focuses on the systemâĂŹs behaviour: what it does, and why. CSE therefore focuses on what it describes as cognitive systems, and how these are comprised of joint cognitive systems.

CSE uses a range of methods for collecting data. These include observation (which may be recorded), interviews (including Knowledge Elicitation techniques, such as the Critical Decision Method), verbal reports (think aloud protocols) and instruments to collect ratings (surveys and questionnaires) and categorisation data (card sorting). There is some overlap with the methods that ethnographers use, but CSE practitioners use these methods to look at specific issues, often over limited time periods, and analyse the data to produce results that can be used to directly inform system design.

The most recent (and stable) incarnation of CSE is described in detail in the two books by its originators, Erik Hollnagel and David Woods (Joint Cognitive Systems: Foundations, and Joint Cognitive Systems: Patterns in Cognitive Systems Engineering), and developments continue to take place in this area, where there is an active global community. There are strong links between CSE and Resilience Engineering, and these are reflected in the overlapping communities of practitioners and academics.

Cognitive Systems

In formal terms, a cognitive system is any system that can modify its behaviour on the basis of experience so as to achieve specific anti-entropic needs. In other words, a cognitive system can control what it does. Under this definition most living organisms and some kinds of machines are cognitive systems. Machines are a subset of the more general class of artefacts.

Organisations can be considered to be cognitive systems, because they can control what they do. They can also be considered as artefacts too, albeit artefacts that have been designed for a particular purpose, even though organisations are of a social, rather than a physical nature.

The technological aspects of cognitive systems are mainly of interest because of how they are used. These systems are invariably embedded in a socio-technical context: people (and organisations, more generally) are involved in designing, building, testing and using cognitive systems. From the CSE practitionerâĂŹs viewpoint all systems are considered to be socio-technical systems. A distinction is drawn, however, between the technological system, in which the technology plays a central role in determining what happens, and the organisational system, in which people play that particular role. CSE practitioners are mostly concerned with applications in complex dynamic domains, such as aviation, industrial process control, healthcare and so on.

Studies of these domains are based around three identifiable, interleaved threads:

- Understanding how people cope with complexity, particularly the complexity that has arisen through advances in technology and socio-technical changes.

- Understanding how artefacts are used, particularly how they have become an inherent part of peopleâĂŹs activities (both work and leisure).

- Understanding how people and artefacts can be described as Joint Cognitive Systems, and hence how they can work together.

Joint Cognitive Systems

The joint cognitive system (JCS) is the basic unit of analysis in CSE, and is used to emphasise the central idea of co-agency. In other words the human and machine have to be considered together, rather than as separate entities linked by human-machine interaction. CSE uses a functional approach to the analysis of systems, focusing on what a JCS does and why, rather than how it does it.

It is important with JCSs (as it is with any system) to define the boundaries clearly, and make them explicit. The boundary of a JCS is determined by the purpose of the JCS, and the focus of the analysis of that JCS.The distinction between a cognitive system and a JCS is clearest at the level of the single individual: a person is a cognitive system, but cannot be a JCS. A person using an artefact, however, is a JCS, and a group of two or more people is a JCS too. In practice, CSE focuses on JCSs which exhibit at least one of the following characteristics:

- **The functioning of the JCS is non-trivial**. In other words, obtaining a response from the JCS requires more than a simple action. In those cases where more complex artefacts are involved, it may be necessary to engage in some form of planning in order to produce a response.

- **The functioning of the artefact is at least partly unpredictable**. This may be attributable to ambiguities in the design, for example, which make it hard to identify what a particular widget is supposed to do, or could be due to a lack of knowledge or training on the part of the users.

- **The artefact itself entails a dynamic process**. In such cases the pace of events and the way that they develop is not driven by the user. The corollary of this is that time is regarded as a scarce resource.

CSE starts with trying to understand the patterns of work, using observation of JCSs. It then uses this understanding to guide the search to identify the things (people, resources and so on) that would be useful and necessary to support the types of work that have been observed. The results can be used as a basis for (innovative) design, in participation with others, to support work and the processes of change.

Retrospective

There are inherent links between the fields (and the communities) of CSE and resilience engineering, at least in part because Erik Hollnagel and David Woods have been involved in both from the earliest days. CSE grew out of the cognitive sciences and human factors, however, whilst resilience engineering grew directly out of safety engineering. So one of CSEâĂŹs aims was to promote a more cognitive-based view of systems, whilst Resilience Engineering was aiming to change the view of safety from the avoidance of failure avoidance to one that emphasises success and learns from that. CSE ideas have been applied in areas such as aviation, and industrial process control, and more recently in healthcare. The research into CSE continues alongside research in Resilience Engineering, and in some cases is being used to provide a foundation for engineering resilience into new systems, particularly in healthcare (and particularly in the USA). This work is likely to continue for the foreseeable future.

Chapter 6

Resilience Engineering

Gordon Baxter, University of St Andrews

Summary

Resilience engineering is concerned with building systems that are resilient to change. In other words systems that continue to work, often through the results of human endeavours, when faced with adverse situations (both anticipated and unanticipated). The work grew out of the safety engineering community around the end of the 20th century. There is a strong relationship between resilience and dependability, with resilience being described as the persistence of the dependability of a system (or organisation) when facing changes. The discipline of resilience engineering focuses on three main areas: developing tools and techniques to assess how organisations achieve resilience in their particular domain; on improving organisational resilience; and on modelling and predicting how organisational change and decision making affects risk and resilience.

Background

Historically, safety engineering has focused on the negative aspects of systems, and tried to achieve failure rates that are as low as reasonably practical (such as 1 failure in every 10,000 events or 10-4). In this view, a system is perceived as being made safer if the number of adverse events is reduced. This approach to safety engineering analyses what goes wrong, looking for failures and malfunctions, and then tries to prevent recurrences by eliminating causes and putting appropriate barriers in place. Resilience engineering was developed to take a more positive view of safety. In the resilience engineering view, safety is regarded as the ability to succeed under varying conditions. Systems succeed, far more often than they fail: a failure rate of 1 in 10,000 events, means a success rate of 9,999 in 10,000 events. The resilience engineering approach therefore analyses why things go right, and uses that as a basis to understand what counts as normal performance, so that work can be made better and safer. Work situations are invariably underspecified and therefore not completely predictable, so resilience engineering looks at issues to do with performance variability which is not only necessary (to deal with the changing situations) but also inevitable because of the inherent variability of people, organisations, contexts and technology.

The Trade-off Between Efficiency and Thoroughness

The Efficiency-Thoroughness Trade-Off (ETTO) Principle was formulated to help explain why things that often go right can sometimes go wrong. It is not really a new principle, it is more a way of integrating lots of similar work together under a single unifying umbrella. Examples of the ETTO appear to be ubiquitous. Efficiency is achieved when a particular goal (or objective) is attained at minimum cost (time, effort, resources and so on); thoroughness involves carrying out a detailed analysis that allows one to be confident that the current conditions will lead to some desired activity being successful and having no unwanted side-effects. When the balance between efficiency and thoroughness is achieved, successful performance results. If the balance tips too much towards efficiency this can lead to wrong actions being performed (through lack of analysis of the situation); if the balance tips too much towards thoroughness this can lead to actions not being performed too late to be effective, because so much time has been spent on analysing the situation. In order to manage safety, it is important to understand how the balance between efficiency and thoroughness is realised. ETTOs can happen for several reasons:

- Scarcity of resources, particularly time, or uncertainty about the amount of time.
- The inherent human trait of following the line of least effort.
- A need to keep something in hand (reserves of resources, or time) to handle unanticipated situations.
- Peer pressures to do things in a particular way or to meet a specific deadline.

- Organisational pressures, such as the conflict between priorities (safety first) and practices (be ready on time).
- Individual characteristics, such as personal priorities, working habits, and personal ambitions.

The trade-off is a heuristic one that applies to people and to organisations. It can only be made by machines when it has been included in their implementation (embedded in the software), and in such cases is algorithmic, rather than heuristic.

The Functional Resonance Analysis Method (FRAM)

The FRAM was developed as a way of describing the performance of socio-technical systems. It regards variability as being inherent in normal performance, and uses this to explain why accidents happen: performance variations can lead to positive as well as negative outcomes. Shortcomings in performance, however, are linked to variability in complex relationships, so they cannot be adequately described using simple linear models. Some adverse events can be attributed to the breakdown in normal functions, but generally they are best understood if considered as the result of the combination of several sources of variability in human performance. The FRAM is built on four principles:

1. **The principle of equivalence of successes and failures**. People and organisations continually have to adapt to the current conditions. When these adjustments are made correctly and when failures and potential harms are correctly anticipated, this leads to success; when this ability to correctly make adjustments is absent, failures can result.
2. **The principle of approximate adjustments**. Work situations are invariably underspecified and hence partly unpredictable. Individuals, groups and organisations have to adjust their performance to suit the prevailing conditions. These adjustments are approximate because resources (time, information etc.) are scarce.
3. **The principle of emergence**. The variability in several functions can combine in unanticipated ways, giving rise to consequences that are disproportionately large, and produce non-linear effects. Performance (successful or otherwise) is emergent, rather than resultant.

4. **The principle of functional resonance**. When the variability of several functions resonates, this can cause the variability of one function to exceed its normal limits. These consequences can dissipate through tight couplings rather than well-defined cause-effect links.

Cause-effect models have traditionally been at the core of safety engineering. These structural approaches (such as MTO: (hu)man, technology, and organisation) can be used to provide analyses of complex situations, but the analyses are necessarily linear because they are based on simple direct relationships (cause-effect). The FRAM focuses on the system dynamics (and variability) rather than modelling individual failures, and hence can be categorised as a systemic analysis approach (like Nancy LevesonâĂŹs Systems-Theoretic Accident Models and Processes, STAMP). These more holistic approaches describe events as coupled functions, with links between functions showing dependencies, rather than cause-effect relations.

Retrospective

The field of resilience engineering is still a relatively new one, and it is continuing to develop. The move towards a more systemic, functional approach to understanding system performance, rather than a structural approach reflects the need to find new ways to deal with the ongoing rise in complexity in systems. Models like the FRAM (and STAMP) appear to offer much promise in this area, and have been successfully used in domains such as healthcare, aviation, and finance. The models continue to be developed as they attract more and more users. As organisations and nations increasingly focus on critical nature of networked (and national) infrastructures, the need for resilience engineering methods and tools continues to grow. There is an active and growing resilience engineering community, centred around MINES, ParisTech (in the south of France), which significantly overlaps with the cognitive systems engineering community. They run regular conferences on resilience engineering, and a training school for use of the FRAM.

Chapter 7

Computer Support for Cooperative Work (CSCW)

John Rooksby, University of St Andrews

Summary

CSCW is a interdisciplinary area of research that examines technology support for social practices such as working together or multiplayer gaming. The acronym refers to Computer Supported Cooperative Work, but this is a legacy term that does not truly reflect the broad interests of the area. The key innovation of CSCW is that it takes primary interest not in technology, but in the kinds of human and social practices that are to be supported by technologies.

Background

The term Computer Supported Cooperative Work (CSCW) was coined in the mid 1980s. At the time, computing equipment was becoming more and more commonplace in workplaces, and the problem of software support for workgroups had emerged as a new research issue. There was much interest in groupware, for example Lotus Notes. However, as time went on, computers kept on becoming more and more affordable, not just to organisations but to general consumers. At the same time, networked computing was becoming the norm, first with local networks, then the Internet, and then wireless and mobile networks. Whereas CSCW started out with a somewhat short-term vision of developing software solutions for workgroups, the terrain in which it was interested opened up massively. The discipline remains very much alive today, often seen as a more socially oriented sibling of the field of Human Computer Interaction.

The Acronym

CSCW stands for Computer Supported Cooperative Work. Understanding what these terms mean are helpful for understanding the field, although many feel the acronym is now out of date:

- **Computer Support**: The original focus on groupware for desktop computers has given way to a broader focus. This includes software support for groups or ensembles of people (for example messaging systems, social network sites, wikis, versioning systems, and so on). It also includes the use of mobile devices, appliances, interactive displays and so on. The concern is still how these fit with and enable human practice, but "support" is not always the right term. For example, video games constitute the practices of games playing rather than support it. Even though "support" may be dated, the orientation of CSCW studies still remains very much to systems in practice.

- **Cooperative**: The second C in CSCW stands for cooperation, not coordination and not collaboration. These terms are often mixed up, and while some do not believe this is important, others argue it is a mistake. The argument is that cooperation can be used to mean collaboration or coordination, but can also mean work that has some impact between parties who are not collaborating, and perhaps not purposefully coordinating their work. This may, for example, be working with shared resources (such as scientists sharing a high-end computer), or it may be shared interests or focuses (for example hospital staff do not necessarily collaborate but must remain coordinated across a hospital and between shifts). It may also refer to interaction between adversaries or rivals (say in a sports game or an auction). Whether or not you think terminology is important, the essential thing is not to assume that CSCW implies a concern solely with groups and collocated action.

- **Work**: The initial focus in CSCW was on office work, but over the years this shifted. The CSCW papers in this handbook focus not only on office work, but on work in hospitals, banks, factories, and so on. They also look beyond work, for example, at technology use in the home. Pointing out that CSCW now looks widely beyond workplaces, Mike Twidale has suggested the discipline become known as CSC*, where the star is a wildcard that could mean any cooperative endeavour (Leisure, Gaming, Education, Tourism, Commuting, etc). Others however contend that the word 'work' is flexible enough to cover any kind of human labour, whether this be working in an office, working on playing a computer game, or working though messages on a social network site. Again, the terminology is not the key issue, the point is that CSCW is concerned generally with human, cooperative practices, and not simply with office work. The term CSCW is, for many, out of date. Indeed the book series published by Springer has chosen to call itself CSCW: Collaboration, Sociality, Computation, and the Web. This is not to say the area is dying, far from it, it has simply outgrown its original concerns.

Interdisciplinary Research

What sets CSCW apart in computing related disciplines is its focus on "work" (or more general human and social practices) rather than technology. That is, research questions and results in CSCW generally discuss the requirements for or effects of technology regarding human practice. Because of this focus, CSCW has become highly interdisciplinary. CSCW draws from technology disciplines as much as those that have historically studied work-practices (for example Psychology, Sociology, Organisational Studies, and so on). How CSCW differs from these other disciplines are its concerns with the details of practice, how things are actually and specifically done, and how technologies, specifically, feature in this. The orientation is to how to design or deploy technology in ways that are supportive of cooperative practices. Therefore it is often essential to look at the details of use rather than broader trends. It is incorrect therefore to assume that any sociology, or any psychology is interesting to CSCW, and incorrect to assume that methods from these disciplines can simply be imported. CSCW is not the sociology or psychology of technology but a melting pot in which technologists, sociologists, psychologists and others have come together to innovate ways of understanding and designing systems. Ethnography is one example, it has taken on a very different form and set of concerns to what many sociologists think of as ethnography.

Common Interests

Most work is cooperative in some way, and the task in CSCW is not so much to separate out cooperative work as one kind among others, but to investigate the cooperative aspects that feature within people's broader efforts and concerns.

Therefore a CSCW researcher may approach one of many settings and seek to uncover how cooperation is or could be done with technology. They may seek to address one of a number of themes, including:

1. **Awareness**: In what ways can one person be aware of the actions of others? This is particularly important when actions can have an effect on other people and may not necessarily be visible. The problem is not just how to increase awareness, but how to do this appropriately (for example without compromising privacy).

2. **Articulation**: In what ways can people's efforts be divided up and kept separate? This is important where there could be clashes or wastage if practices begin to consume the same resources or are unnecessarily repetitive.

3. **Plans and Action**: How are plans achieved in actual, situated practice? Plans, including project plans, workflows, protocols and so on do not determine how work is done, but rather it is an achievement of that work to have gone to plan.

4. **Timing**: There are a variety of issues associated with the temporal features of work. How can cooperative work be effectively scheduled? How can information be delivered in timely ways? What are the patterns and routines that people engage in? Ultimately, how can technologies support timely work and do things in a timely way?

5. **Ordering**: How can cooperative activities be done in order? What role can technology play in ensuring that one thing is done after another, rather than simultaneously, or not at all? Conversely in what ways does technology impose order inappropriately?

6. **Interaction**: What role does technology play in the ways people interact, and how does human interaction take place through and around technology? How do people talk as they work, and what do they talk about? How does communication technology play a role in wider forms of communication (for example how and when do people talk about emails they have sent, or chose to use IRC rather than email?)

7. **Leadership and management**: How can technologies be used to appropriately monitor and manage cooperative work? How can leaders (as opposed to managers) make better use of technology?

8. **Power and Politics**: How do technologies affect the distribution of power within workplaces? How can technologies be used to empower people, improve social settings, and so on?

The above is not an exhaustive list, but sketches out some frequent interests.

Technology 'in the wild'

CSCW researchers often find a great deal of interest in the settings in which technologies are used. CSCW research is often done in 'the wild' because the complexities of the settings in which work is done and the agendas and concerns they need to satisfice can be difficult to predict and simulate in controlled settings. Key points for doing this include:

- **Descriptions of work often differ to actual work**: People often describe their work in normative and rational ways, whereas the realities can be somewhat different. In particular many of the details of exactly how a technology is used may not be seen as interesting or remarkable by the user but can be of profound interest to the researcher.

- **Multiple technologies are often in use**: Most cooperative practices take place across a variety of technologies (e.g. programmers cooperate across code, versioning systems, plans, email, talk, IRC and so on). Often, any new technology has to be seen in the context of others.

- **Extra artefacts or technologies are often in place**: The users of technology often configure other technologies around them. So all-in-one software solutions may in reality be supported by the use of spreadsheets or other software. Complex technologies may be supported by the use of cheat-sheets, and so on.

- **Technologies are used in unintended ways**: Technologies may not always be used in the ways they were designed. A classic example is that people may write in the margins of paper forms. If these uses are not understood when developing new technology (and either supported, or the underlying causes eliminated) then problems will arise.

- **The layout of a setting can have effects on interaction**: The context in which a technology is used will affect its use. In co-present settings people may do things like ask for help rather than read documentation. If a computer is situated too far from where an action takes place (e.g. away from a bed in a hospital) then this can affect the ways record keeping is done. And so on.

Again, the above is a flavour rather than an exhaustive list. Because CSCW is focused upon practice, because people routinely make use of multiple technologies as a part of their practices, and because other aspects of the setting in which that practice takes place, CSCW researchers are often interested in what can be described as 'systems ecologies'. Ecology is a useful metaphor because it refers to the interdependencies of elements of an environment without necessarily implying stability or harmony. In particular it implies that introducing something new will potentially have wide-ranging effects that are difficult (or perhaps impossible) to predict, and so important to monitor.

Retrospective

CSCW provides a melting pot for ideas from several disciplines interested in technology and human practices. This has provided a context in which methods and perspectives relevant to engineering socio-technical systems have thrived. However the corollary of this has been something of a lack of a clear focus and an importing of longstanding disputes from other disciplines.

Moreover, it can often seem that little progress is made in CSCW. The problems that CSCW grapple with are tameable but not resolvable; there will be no once-and-for-all solution to the problems of cooperative work. As technologies move on, many of the issues prove to be the same, with the same insights confirmed for new settings. The focus in CSCW on the human practices, rather than on producing novel technology, often also means that it provides a commentary on new technology rather than provides an arena for its invention. This is not necessarily a bad thing, as the management and deployment of technology is often more of a challenge than its production. Many organisations understand that the real costs of software are not in its shelf price but in its long-term use.

Chapter 8

Fieldwork, Ethnography and Ethnomethodology

Mark Rouncefield, Lancaster University

Summary

Driven by the 'failure' of systems that manifestly did not meet the needs of their users, fieldwork is an approach to the study of work where an observer engages directly with work in its own environment, with a view to understanding the 'real' processes, activities and interactions of the people involved. Ethnography is an observational approach that examines work as it is practised in a naturalistic setting and ethnomethodology is an approach to analysis that gives precedence to the actors their ways of structuring work rather than attempting to analyse this using some theoretical framework.

Fieldwork

If design is more of an art than a science, dealing with messy indeterminate situations and 'wicked problems', then before designers can solve a design problem they need to understand some basics - such as what they are designing, what it should do and who should use it and in what circumstances. So methods needed to be more attuned to gathering relevant data in âĂŸreal worldâĂŹ environments; that is, the social settings in which systems were likely to be used rather than in laboratories.

Fieldwork is an approach to research and the collection of data that involves actually going to site where a system is being used, or where a new system is proposed, in order to study the natural circumstances of work and activity that any system is designed to support. The method is an alternative and response to the perceived weaknesses of those experimental methods that seek to replicate features of the setting in the laboratory.

Ethnography

Ethnography is a qualitative orientation to research, derived from anthropology, that emphasises the detailed observation of people in naturally occurring settings. The fieldworker experiences the environment in the same way as the people in that environment and observes their activities and interactions. The move towards naturalistic observational methods in anthropology is generally attributed to Malinowski and the conviction that only through living with and experiencing 'native' life in their own environment could a researcher really understand that culture and way of life. In CSCW and HCI, the ethnographic move is strongly associated with Lucy SuchmanâĂŹs âĂŸPlans and Situated ActionsâĂŹ and the Lancaster ethnographies of Air Traffic Control conducted by Richard Harper and Dave Randall.

The main virtue of ethnography is its ability to make visible the âĂŸreal worldâĂŹ sociality of a setting producing detailed descriptions of the âĂŸworkadayâĂŹ activities of social actors within specific contexts. It is a naturalistic method that seeks to present a portrait of life as seen and understood by those who live and work within the domain concerned. It is this objective which is the rationale behind the methodâĂŹs insistence on the development of an âĂŸappreciative stanceâĂŹ through the direct involvement of the researcher in the setting under investigation. In the study of socio-technical systems ethnography has primarily focused upon the study of work and settings for which new technology is being designed with the intention of informing that design.

Ethnography has acquired some prominence as a fieldwork method that could contribute both to a general understanding of systems in use in a variety of contexts and to the design of distributed and shared systems. Efforts to incorporate ethnography into the system design process have had much to do with the (belated) realisation, mainly among system designers, that the success of design has much to do, though in complex ways, with the social context of system use. A number of well publicised âĂŸdisastersâĂŹ (The London Ambulance System) suggested that traditional methods of requirements elicitation were inadequate, or in need of supplementation, by methods better designed to bring out the socially organised character of work settings. Ethnography with its emphasis on the in situ observation of interactions within their natural settings seemed eminently suited to bringing a social perspective to bear on system design.

This âĂŸturn to the socialâĂŹ in design arose out of dissatisfaction with existing methods as offering overly abstract and simplistic analyses of the nature of social life. Additionally, this 'turn to the social' recognised a new kind of end-user, a âĂŸreal time, real worldâĂŹ human being and consequently designers turned to the social sciences to provide them with some insights, some sensitivities, to inform design. Requirements elicitation has to be informed by an analysis of the âĂŸreal worldâĂŹ circumstances of work and its organisation. The virtue of ethnographic approaches comes from this recognition that computers are enmeshed into a system of working and incorporated in highly particular ways - used, misused, modified, circumvented, rejected - in the flow of work. One of the virtues of ethnography lies in revealing these myriad usages in the context of âĂŸreal worldâĂŹ work settings seeking to answer what might be regarded as the essential socio-technical question - what to automate and what to leave to human skill and experience?

Ethnomethodologically informed ethnography

An ethnographic stance entails viewing the social world from the standpoint of its participants - data is collected about their everyday actions and interactions. One collected, a detailed analysis of this data takes place. This analysis of ethnographic data can be attempted from a number of different analytic perspectives âĂȘ such as Activity Theory, Distributed Cognition, Actor-Network Theory and so on. In ethnomethodologically informed ethnographic research the understanding of any setting is derived from the study of that setting itself, rather than from any highly structured model or theory of work organisation or work processes; it ties itself closely to the observed data, it is 'data-driven'. The approach recognises the inherent 'messiness' of the world and the inadequacy of any theory to deal with this.

An ethnomethodological approach is an atheoretic approach to this analysiswhere a member's methods for accomplishing situations in and through the use of local rationalities become the topic of enquiry. For ethnomethodologically informed ethnographic enquiry, the people in an environment, their subjective orientations and experiences are central. Observation focuses on the places and circumstances where meanings and courses of action are constructed, maintained, used and negotiated. A central precept of ethnomethodological ethnography is to find the orderliness of ordinary activities, an orderliness accomplished by social actors, unreflectively taken-for-granted by them and constructed with their common-sense knowledge of social order.

In acknowledging the âĂŸsituatedâĂŹ character of work, ethnomethodologically informed ethnography displays how even in the most apparently routine activities workers need to use their judgement and discretion in response to the various contingencies that arise. In consequence, the accomplishment of work tasks involves a range of tacit skills and local knowledge that may be rendered invisible by formal models of processes or procedures, often going unrecognised by the workers themselves. In ethnomethodologically informed ethnography, the phenomena which are to be investigated are to be studied in their character as âĂŸphenomena of everyday lifeâĂŹ as âĂŸeverydayâĂŹ occurrences for those who are involved in the activities in question, and the investigator is, therefore, seeking to ascertain what the phenomena mean for them.

In studies of the kind that ethnomethodologically motivated ethnographers make, the concern is with the depiction of âĂŸthe working sensibilityâĂŹ of those under study. The interest is remote from the kinds of general reflections that someone in an occupation can produce, and much more engaged with their consciousness and attention when they are âĂŸat workâĂŹ: what kinds of things do they take for granted or presuppose in going about their work, what kinds of things do they routinely notice, what kinds of things are they âĂŸon the lookoutâĂŹ for, how do they âĂŸtune themselves inâĂŹ to the state of being âĂŸat workâĂŹ, what are the constituents of their âĂŸserious frame of mindâĂŹ, how do they react to the things that occur within their sphere of attention, what objectives are they seeking to attain in their reactions to whatever occurs, and by what means - through what operations - will they seek to accomplish those objectives in adaptation to these unfolding circumstances. Thus, attention is focused on the study of doing the work. The emphasis is on work in the raw, work as it is done, and in the ways in which it is done in actual practice, as opposed to work in idealised form as presented in organograms and process models.

Retrospective

Ethnographies of work have proven to be very useful in developing an understanding of how work is practised and hence in informing systems engineering processes of specification and design. They are particularly effective in settings, such as control rools, where the people involved are co-located and the work involves coordination of different people. For distributed work, conventional ethnography is more difficult and expensive and so we have much less experience of the contribution that it can make.

Most ethnography has focused on the 'users' of a system and hence the information derives is most effective in understanding user needs and constraints. Fieldwork is also valuable in understanding broader organisational needs but observations must be supplemented by other techniques such as interviewing and document analysis to understand the 'organisational' requirements and constraints on systems design.

Chapter 9

Ethnographic Viewpoints

Introduction

We have discussed elsewhere in this handbook the use of ethnography as a fieldwork method that is used to gather information about how work is actually done, as distinct from the abstract models of work that may be presented by an organization. Ethnographic studies build a rich picture of work that includes information about cooperation in the workplace, how people cope with problems, how representations are used to support work and so on.

Ethnography developed in the social sciences and the fieldwork record is typically a narrative document that can be used as a basis for extensive post-study analysis . However, unstructured narrative is not readily accessible to anyone apart from the ethnographer himself or herself and, in particular, it is not a helpful document for engineers involved in complex software development.

To address this problem, we have investigated how ideas from requirements engineering may be applied to structure and present the information in the fieldwork record. In essence, we propose a number of ethnographic viewpoints which collect together related information from the fieldwork records.

Viewpoints

The notion of a viewpoint originated with the idea that different stakeholders in a system see that system in different ways - they have their own 'viewpoint'. and A number of requirements engineering methods were developed in the 1990s based around this idea, including work by Finkelstein and Nusiebeh at Imperial College and at Lancaster University by Sommerville, Kotonya and Sawyer.

As this work developed, the notion of a viewpoint evolved from a stakeholder perspective to a structuring device that was used to organize and present a related set of requirements. So, in a medical records system, there might be Doctor, Nurse, Administrator and Manager viewpoints. These do not just represent a single perspective e.g. what is required by nurses but rather they collect together all requirements that are related to nurses. These requirement may not come from the nurses themselves but may come from other stakeholders, such as administrators, who have expectations about what nurses will do.

Viewpoints have a number of benefits:

1. They are useful in organizing requirements, especially when presenting these to different classes of stakeholder. Therefore, when presenting to nurses, the focus would be on the requirements from the Nurse viewpoint with less emphasis given to other viewpoints.

2. They provide a basis for requirements elicitation - therefore, if viewpoints are associated with a class of stakeholders, these are the primary sources of information. In other cases, the viewpoints are more abstract but represent a checklist of areas which should be covered in the requirements elicitation activity. When viewpoints are explicitly identified in one system, they may be reused in the elicitation of requirements for later systems.

3. They support requirements analysis in that related requirements in different viewpoints can be compared for overlap and conflicts. These related requirements typically arise where different types of stakeholder make use of the same system functionality.

The paper by Sommerville and Sawyer 'Viewpoints: principles, problems and a practical approach to requirements engineering' summarises the use of viewpoints in requirements engineering.

Our experience with the use of viewpoints in requirements engineering prompted us to think about how these could be used to support the ethnographic process and provide some structure for ethnographic analysis.

Ethnographic viewpoints

Primarily, ethnographic viewpoints are a means of organizing the ethnographic record and presenting this back to the actors in the field site that have been observed. The motivation for the development of these viewpoints was the need to share the ethnographic record between the fieldworker who collected the data and the system engineers who required that data to inform the system requirements and design.

With experience, we discovered that these ethnographic viewpoints could be used by people who were not experienced ethnographers to guide and organize the ethnographic process itself. This is discussed elsewhere in this handbook in the section on the Coherence method.

There are three principal ethnographic viewpoints:

1. A work setting viewpoint, which focuses on the place where the work is done and how this is organized to support the work.

2. A social and organizational viewpoints, which is concerned with the interactions between the people involved in the work and how they cooperate. It also includes how the people in the workplace are influenced and affected by organizational issues such as organizational structure and policies.

3. A work flow viewpoint which presents information about the sequence of work activities, the representations used at each stage in these activities, etc.

The work setting viewpoint

The first of our viewpoints focuses on the development of a representation of the setting of work and how users work within a flexible working division of labour.

One way of reflecting the practical everyday nature of the work is for accounts of the fieldwork to focus on its setting. This is often reported in terms of the physical layout of the location in which work is taking place. This viewpoint seeks to represent the spatial distribution of the work place in terms of its participants, the work they do and the local resources that they use. The purpose of this is to provide a sense of 'where the work takes place' and the socially constructed affordances that this offers as an arena of various kinds of interactions that take place. In this respect, it is a view upon the workaday character of the world within its setting.

The viewpoint was originally developed after experience with ethnographic studies of co-located work e.g. a control room, where many people were cooperating. When we apply this to distributed work, we are no just interested in the individual workspaces but also in the tools used by the actors to support their work and, particularly, how they organize the technology in their workspace to support interaction and collaboration.

Social and organizational perspectives

This viewpoint highlights the loosely structured and practical focus of ethnographic studies of work. The emphasis of ethnographic studies is on providing materials on the 'real world, real time', nature of work which can be used for later analysis. These materials furnish portraits of the practical nature of work, often presented as illustrative vignettes within a larger report. The analysis, again typically, tries to bring out the day-to-day experience of the work from the point of view of various actors within the setting. Each of these actors have informal incomplete and often inconsistent models of the work taking place each of which provides significant insight in the development of abstract models in the requirements process.

This viewpoint aims to collate summaries of this body of observational information in such a manner that they are accessible by developers as a resource for system requirements specification.

Given the relatively discursive presentation of this material as well as its focus on the subtle, often 'invisible', often tacit, features of workaday activities, this viewpoint is really a collection of potential viewpoints from which such materials can be examined, depending upon the interests of the designers. Observers and developers are free to add additional perspectives relevant to the study as a means presenting this information. For example, it may be presented from the point of view of a particular actor within the work setting, a sequence of tasks, a collaborative endeavour, the policy of the organisation, and so on.

This viewpoint should also take into account the collaboration tools that are used to support distributed work. Email is, of course, now ubiquitous and is still probably the most commonly used collaboration medium. However, synchronous collaboration (Skype, phone conferencing, instant messaging) may also be used as well as social networks such as Facebook and Google Plus.

Work flow

This viewpoint, again, is probably best seen as a collection of potential viewpoints, focuses more directly on sequences of work activities, information flows, and so on. In this respect it emphasises and exhibits the division of labour within the work along with its various interdependencies; interdependencies, it is important to stress, which are not always ormally specified. The kind of fieldwork materials germane to this viewpoint include 'tracking work' through its sequences and transformations, such as a particular piece of software through error testing, the flight of particular aircraft through UK airspace, invoice processing, etc..

Once again, such materials will consist of reports of activities, the relationships among parties to the work, how the interdependencies are achieved as 'real world, real time' phenomena, the contingencies that can arise, how they are dealt with, and so on. In documenting workflow, it may be possible to use one of the many diagrammatic notations that have been developed in this area such as UML activity diagrams. However, these have been developed from a particular model of work based on activities, inputs and outputs and this is not appropriate for documenting all types of workflow.

As part of this viewpoint, it is sometimes useful to examine how the work is represented and how this representation is used by the different actors involved. For example, in work such as invoice processing, a paper or electronic invoice may be annotated with comments by people at each stage of its processing.

Chapter 10

Coherence

Ian Sommerville, University of St Andrews

Summary

Coherence is a systematic approach to carrying out field studies of work. It suggests that observations of work should be focused around three social viewpoints namely distributed coordination, plans and procedures and awareness of work as well as a number of cross-cutting social concerns namely paperwork and computer work, skill and the use of local knowledge, spatial and temporal organisation, and organizational memory. The UML should be used, wherever appropriate, to document fieldwork.

The chapter on Requirements and Design discusses more general issues of fieldwork and requirements engineering.

Background

The work on Coherence was proposed to provide guidance and a framework for people without experience or training of ethnographic fieldwork to carry out observations of work, organise the record of these observations and document them using a notation that is familiar to software engineers. Realistically, few organisations have access to trained ethnographers and the intention of Coherence is to allow software engineers to be directly involved in fieldwork and communicate their results to colleagues in a familiar notation.

The Coherence approach is primarily intended to support the process of requirements engineering and a possible outcome of applying the approach is a set of use-cases that can be a starting point for more detailed analysis and design.

The starting point for the design of Coherence was PreView, a method for requirements engineering that incorporated the notion of viewpoints âĂŞ different perspectives on the requirements and concerns. Concerns are cross-cutting system wide notions (such as reliability) that are relevant to all viewpoints. The PreView approach is based on decomposing concerns to questions then putting these questions to stakeholders to understand their requirements and constraints. Papers on PreView are included in the Appendix to this chapter.

Social viewpoints

Ideally, merging all of the requirements from all viewpoints should give you all of the requirements for a system but, in practice, this is rarely the case. There are usually system-wide requirements which are not associated with any viewpoint and which are often constraints or âĂŸshall notâĂŹ requirements (the system shall not allow access to unauthorised users e.g.). As we shall see, these system wide requirements are derived from so-called concerns.

The idea of social viewpoints is that these viewpoints capture the requirements that emerge from the notion that work is a social process that is embedded in an organisation with its own culture and ways of working. The social viewpoints that we propose come from our experience of fieldwork in a range of settings, where these viewpoints have been relevant in different places. The three social viewpoints in Coherence are:

1. **Distributed coordination** âĂŞ the ways in which a team of people, who may be working in distributed locations, organise their work to allow it to be coordinated.

2. **Plans and procedures** âĂŞ the formal model of work and the way that this is manifested in formal objects in the workplace. Also of interest is the ways in which work as it is practised, deviates from these formal expressions of the work.

3. **Awareness of work** âĂŞ the ways in which people in the workplace make others aware of what they are doing and how they themselves become aware of the actions of others.

Social concerns

Concerns are cross-cutting notions that are applicable to all viewpoints (and to the system as a whole). The simplest way to understand what a concern is to think about characteristics of the system as a whole âĂŞ such as reliability, safety, etc. (which is where PreView started). Social concerns are analysis issues that affect all viewpoints:

1. **Paperwork and computer work** âĂŞ how are existing paper-based and computer-based systems used?
2. **Skill and the use of local knowledge** âĂŞ to what extent is the operation of the system dependent on the skills and knowledge of individuals who are part of that system.
3. **Spatial and temporal organisation** âĂŞ how is work organised in physical space and time?
4. **Organizational memory** âĂŞ how are plans and procedures and coping strategies for errors remembered in an organisation

The key elements of these concerns is that they are the starting point for more detailed decomposition which eventually leads to a set of questions that have to be answered either by questioning the appropriate system stakeholders, from the system documentation or from observations of people using the system.

Question-driven analysis

Viewpoints and concerns steer the analysts attention towards social issues that we know are often important in analysing organisational systems. To gather information that is relevant to these social viewpoints and concerns, we have extended the question-driven approach in PreView. Generic and specific questions are identified and the answers to these questions provide information that can be the basis for identifying system requirements and associated UML use-cases.

Each viewpoint has an associated set of focus questions, which are explained in more detail in the papers included in the Appendix to this chapter. Examples of focus questions are:

1. **Distributed coordination**: How clear are the boundaries between one personâĂŹs responsibilities and anotherâĂŹs?

2. **Plans and procedures**: What happens when formal plans and procedures fail?

3. **Awareness of work**: How does the spatial organization of the workplace facilitate interaction between workers and with the objects that they use?

The analysis of the social concerns is also question-based. In this case, you start with a concern âĂŞ say Paperwork and Computer Work and decompose this into sub-concerns âĂŞ say, Use of paper, Use of web, Use of local files and, if necessary sub-sub concerns. You then identify a set of questions for each sub-concern that help gather the information you need. For example, for the Use of Web sub-concern, questions might be:

1. To what extent do users routinely consult web sites for information?

2. Are there âĂŸtrustedâĂŹ web sites that are frequently used?

3. How do users share information about trusted web sites?

4. Are internal web sites used?

How you find the answers to these questions depends on the work being studied. Sometimes you consult documents, sometimes observe what people are actually doing and sometimes you can ask them directly. If you ask questions, howeer, you should check by observation that what people actually do is the same as what they say they do.

UML representation

The idea underlying Coherence was to use the UML, wherever appropriate, to represent the work being studied âĂŞ so the UML models that are produced were models of the system as it is, rather than models of the system that is required. The three UML diagram types that are most useful are:

1. **Use cases**, to identify specific work activities

2. **Sequence diagrams**, to show the order of activities/sub-activities

3. **Object diagrams**, to represent objects in the workspace

The problem with UML modelling (which is a generic problem rather than specific to this case) is that the UML is really not good for modelling exceptions where the ways that the exception is handled depends on when it happens, where it happens and who is available. We advocate simply using diagram annotations to handle exceptions rather than trying to create lots of exception use cases or to use conditional sequences, etc.

When we invented Coherence, we anticipated (correctly) that the UML would become the standard modelling notation but over-estimated the impact of the UML on practical software engineering. The advent of agile methods and minimal documentation has meant that many small to medium sized development projects donâĂŹt develop system models.

Therefore, if you use Coherence as a framework for helping you understand the social nature of work, you may prefer to document your fieldwork in a less formal way, which can then be used in discussions about the system requirements.

Retrospective

We believe that the original ideas behind Coherence are still relevant and that. the general problem of providing help and guidance to people who need to understand how social and organisational issues affect work remains.

Coherence was developed before the widespread use of the WWW to support work and, without doubt, could and should be evolved to take this into account. To be effectively used, tool support is probably necessary and whilst prototype tools were developed in a separate project for PreView, these were never extended to take social viewpoints and concerns into account.

Coherence provides a general framework for a process of social analysis but does not, in itself, outline a process that could be used by software engineers. There is a need for the Coherence approach to be developed so that much more guidance is provided for people who are getting started with the process. In particular, the distinctions in practice between social viewpoints and social concerns have to be clarified. The approach should also be revised to take recent development in the UML into account.

Chapter 11

Patterns of Cooperative Interaction

David Martin, Xerox Research Centre, Grenoble

Summary

Patterns of Cooperative Interaction was a project we conducted in order to try to structure a series of findings from different ethnographic fieldwork studies. We created a uniform pattern template which named and discussed a regular pattern of work arrangement and activities found in various fieldwork studies then described the various individual variants we had found in different studies, noting their core similarities but also their differences. The patterns collection is intended to:

- provide a common language for talking about different fieldwork findings across different settings
- give a means of comparing and contrasting similar work arrangements and practices within different settings, and
- enable a variety of practitioners to access fieldwork results easily and to have an introduction to the analytic approach of fieldwork.

Importantly, as well as having these pedagogic features, the patterns can also be used alongside requirements analysis as a means of leveraging analysis in a novel site. The idea is that practitioners interested in social analysis can see whether any activities, arrangements or technologies in the new site are similar to those

discussed in any of the patterns. This can aid them in thinking about important activities to support in any redesign and whether discussed technology solutions might be appropriate.

Background

After the first ethnographic field studies in the computing domain in the late 1980s during the following decade there was a steady stream of them âĂŞ from control room studies, to industrial settings, to offices, finance and banking, looking at cooperative applications, video conferencing and even moving on to leisure and gaming. This pointed to the success of the confluence of computer science and sociologists, especially in the fields of Computer Supported Cooperative Work (CSCW) and Human Computer Interaction (HCI). However, various complaints were levelled at field studies research âĂŞ it was just a series of individual studies with little to connect them, relevant only to the individual sites studied, and in a related fashion, field studies did not produce theories with predictive power.

Furthermore the studies did not tell you what to design to bring project success. It is very arguable that a number of these criticisms could be applied broadly to many approaches âĂŞ there is no silver bullet âĂŞ even theories and approaches that promise prediction or a suggest a method to guarantee results do not in themselves do this. Design is often more of a craft discipline âĂŞ yes dependent on method âĂŞ but often very dependent on personnel and their ability to understand who and what they are designing for, and to work diligently and make good calls in a constrained and contingent setting. However, this did not cover all the criticism of ethnography. While it was legitimate to claim that individual field studies showed their value in themselves it was also clear that there had been few attempts to look more broadly across the corpus and to see in what ways it added up to more than a collection of studies united solely by approach.

Problematically for this project, many of the field studies conducted in CSCW and HCI were âĂŸethnomethodologically-informedâĂŹ; an approach that is a-theoretical and seeks to understand settings as they are locally organised according to how they are understood by participants. Understanding settings in their own terms precludes taking theory and applying it across the board. Hence any attempt to look more generally across settings needed to compare and contrast features in ways that were still faithful to local particularities or else the specific features that made ethnography appealing to designers âĂŞ its in-depth understandings of work âĂŞ would be lost.

Pragmatically, however, it was clear that within individual research articles people did re-use concepts such as awareness (of work between actors) and articulation work (the work to coordinate the work across actors) and this was useful when local instantiations were honoured and the local findings could be used comparatively. But there was just a need to undertake this project more systematically.

Around this time (late 1990s) there had been a flurry of interest within computing about the work of the architect Christopher Alexander. Alexander had produced a magnum opus of theory linked to architectural design patterns about âĂŸdesigns that workedâĂŹ functionally and aesthetically. These patterns had names, descriptions of features and of the âĂŸproblemsâĂŹ they dealt with and how they resolved âĂŸforcesâĂŹ as well as a diagrammatic form and even photograph. They nested within each other according to scale âĂŞ neighbourhood or park within it, house or bedroom. There was something interesting here if also slightly strange and idealistic.

Patterns had been taken up in computing where there had been successful books that collected together software engineering patterns âĂŞ elegant designs for recurrent problems in development. Patterns as an idea were spreading through computer science in general. Consequently we undertook a project to see if we could marry elements of the idea of patterns with the idea that we wanted to try and build connections in the corpus of ethnographic studies. The idea of naming patterns of work âĂŞ or as it became âĂŸPatterns of Cooperative InteractionâĂŹ âĂŞ seemed appealing in terms of describing the arrangements of people and technologies, the activities they carried out, problems and solutions, diagrams and so forth. So the idea became one to extract patterns where we had seen similar phenomena in different settings and design a pattern template to present them.

Pattern Structure

In beginning the patterns project, our first goal was to look for findings or phenomena across settings that bore close similarities. A considerable amount of work was undertaken simply to go through papers in the corpus of field studies. One of the difficulties of this became apparent early on âĂŞ often in the published studies all one had was the fieldwork material selectively cut, prepared and presented for the purpose of making a particular argument. Of course this is what paper writing is about but it made the comparative project harder and we needed to focus largely on our own materials or where we had access to several papers or data, allowing for further analysis.

However, we did find some good candidate patterns so we developed our pattern template that would be loose enough to accommodate the different patterns and would crucially focus on recurrent elements of cooperative work that ethnographic studies dealt with. For example, the group of people and resources used, and the practices they used to coordinate their work. The template is as follows:

1. **Name**: captures the central idea of the pattern
2. **Cooperative Arrangement**: details the actors, resources (artefacts, communication media) involved in the activity described in the pattern.
3. **Representation of Activity**: how the activity is represented (e.g. on an artefact or plan) and the relationship between the two (the activity and the representation).
4. **Ecological Arrangement**: features in the layout of the setting and artefacts and their affordances for the accomplishment of work. Can also have pictorial representation
5. **Coordination Techniques**: practices employed in carrying out action/interaction and how coordination is achieved.
6. **Community of Use**: the user groups or affiliation of actor's involved.

Pattern List and Website Development

In the end we developed a list of ten patterns, and for each we had two or three instantiations of how they played out in different field studies. Each instantiation was described according to the template above. We decided to turn the patterns into a web-based resource. We also added a âĂŸfront pageâĂŹ (literally in the case of the web pages) to each pattern. For the front page a high level description of the phenomena is provided under the heading âĂŸthe essence of the patternâĂŹ. Below this are three more sections entitled âĂŸwhy useful?âĂŹ âĂŸwhere used?âĂŹ and âĂŸdesign implications?âĂŹ. âĂŸWhere usefulâĂŹ details why we have chosen to draw attention to the pattern. âĂŸWhere usedâĂŹ details the two or more specific fieldwork settings we have found examples of the pattern in, and also some brief remarks on similarities and differences between the settings. âĂŸDesign implicationsâĂŹ is used to make some comments about what the identification of the pattern may mean for certain questions concerning 'good', usable, dependable design.

The ten patterns are as follows:

1. Artefact as an audit trail
2. Multiple representations of information
3. Public artefact
4. Accounting for an unseen artefact
5. Working with Interruptions
6. Collaboration in Small Groups
7. Receptionist as a hub
8. Doing a walkabout
9. Overlapping Responsibilities
10. Assistance Through Experience

To give a brief flavour of what the patterns contain we can take the first âĂŞ artefact as an audit trail âĂŞ and summarise as follows.

> In essence this pattern is concerned with the way in which an artefact can serve as a stratified record of work. Amendments and attachments to the artefact, such as comments, date stamps, post-it notes, other documents and so forth readily provide information to actors about the process through which the artefact has progressed in the workplace, seeing who has carried out work, when and why. In this way the artefact serves as a means of coordination between workers.

In this case the pattern was drawn from two studies âĂŞ one of the use of paper flight strips in air traffic control, and the other of the movement of an invoice around an office in a catering firm. We discussed this pattern because both artefacts physically showed what had happened to them, in terms of the work carried out on them during a process. This was recoverable to those in that setting, just like an audit trail, and furthermore, their material nature and placement within the settings could serve as a means for understanding the status of work. This has clear design implications when one thinks that these visible features may be lost in movements to more electronic solutions.

Retrospective: An Unfinishable Project

Overall the patterns of cooperative interaction project was a successful one, however, the more optimistic side of the project was not reachable probably for very understandable reasons. We had produced a patterns resource of reasonable size and managed to place different findings together in a manner that was both faithful to the similarities but also the particularities of the instantiations. The patterns did provide an introduction to some of the findings across ethnographies and the analytic sensibilities of the approach âĂŞ so certainly could be pedagogically useful for a range of practitioners interested in social analysis.

We had also shown âĂŞ how at least in our hands âĂŞ the patterns could aid in the analysis of a novel setting. However, this also pointed to a problem âĂŞ we were experienced with ethnography and field work and therefore could deploy the patterns skilfully, using them in tandem with our requisite knowledge. And although quite a number of people were interested in reading our work and viewing the collection, and drawing on it in a number of ways, we are not aware of anyone else taking it up for analysing a novel setting.

Furthermore, although we spoke with other researchers undertaking similar tasks we were never able to get anyone to contribute further to our collection. This might have been an ownership issue. People would rather start a collection than contribute to someone elseâĂŹs unless it is really successful, but it is also possible that our collection was not really going to be generative in this way. Reasons for this could be that cooperative work seems a bit less inducive to the patterns idea than architecture; our solution and template were always a bit of a forcing device; and it was actually harder to find recurrent patterns across sites, particularly without access to richer fieldwork data. Nevertheless we believe that the patterns collection is useful for those wanting an introduction to ethnographic findings and sensibilities.

Chapter 12

PERE (REAIMS)

PERE (Process Evaluation in Requirements Engineering) is a structured method for analysing processes for weaknesses and proposing process improvements against them. PERE combines two complementary viewpoints within its process evaluation approach. Firstly, a classical engineering analysis is used for process modelling and generic process weakness identification. This initial analysis is fed into the second analysis phase, in which those process components that are primarily composed of human activity, their interconnections and organisational context are subject to a systematic human factors analysis.

PERE is an integrated process improvement method that combines two complementary viewpoints onto the process under analysis:

1. Mechanistic viewpoint–an analysis of the process in mechanistic terms, as a number of interconnected process components. This analysis uses techniques adopted from classical safety analysis, adapted for a consideration of the RE process.

2. Human factors viewpoint–an analysis based on the application of human factors and social scientific principles to assess weaknesses and protections at an individual, group and organisational level using the results of the mechanistic viewpoint to scope the analysis

This dual viewpoint approach has been adopted since it has the following advantages:

1. Structured, usable approach–PERE enables human factors considerations to be presented in a usable manner, through the application of a structured grounded checklist. This checklist is grounded in that each item contains

references to human factors review documents and structured since the user is guided through the checklist by means of navigational questions. This navigation is guided and scoped by the results of the mechanistic viewpoint analysis. As a result, a manageable subset of the checklist is used, preventing the combinatorial explosion of having to consider each checklist item for each component.

2. Sensitive to actual RE process improvement needs–since RE processes in practice combine human and automated processes, it is appropriate to combine two complementary viewpoints within the method, each concentrating on different aspects of the process. PERE exists within the process improvement paradigm and combines both âĂIJhardâĂİ and âĂIJsoftâĂİ process improvement approaches.

3. Knowledge dissemination–PERE integrates classical engineering analysis and human factors analysis. This structured, usable, yet technically defensible approach means that engineers in the process and safety domains will have access to the relevant social scientific research and broader human aspects that determine process dependability and which would not typically be within their domain.

4. Enhanced coverage–since each viewpoint comes from a different research tradition, there is a certain amount of redundancy in the PERE process, resulting in increased coverage of the process under analysis as process weaknesses are trapped under different guises. This redundancy further improves the dependability of the PERE process itself.

Mechanistic viewpoint

PERE's mechanistic viewpoint has its origins in the classical safety analysis technique, Hazops, and Object-Oriented inspired analysis.

For this viewpoint it is assumed that both human and machine activity in the process are analysable into components. The model we describe is based on the principles of using modularity and abstraction to describe systems, considering generic component classes (process, transduce, channel, store and control) as subject to generic component weaknesses, and explicitly considering the "working material".

Once the process structure and working material is described, the PERE analyst completes a PERE component table to describe the process model. This process model is then reviewed for weaknesses by considering the generic weaknesses associated with each component and also the specific weaknesses associated with the components attributes.

In documenting this analysis a PERE Weakness Table is completed. The weaknesses identification and review steps are iterated until no more weaknesses are identified. The results of the mechanistic analysis are then passed on to the human factors viewpoint, although provisional results may be fed forward if, say, one component is considered to be particularly vulnerable to human error.

Human factors viewpoint

In this phase we consider those components that are composed primarily of human activity, their interconnections and working material, and organisational context. The analysis proceeds by means of a series of structured questions, which enables the analyst to search for only those human factor weaknesses that are relevant for the particular process under consideration (e.g. it is not generally necessary to consider knowledge-based component weaknesses for a skill-based component such as typing).

The application of the human factors viewpoint concludes with a completed PERE human factors table, which includes suggested protections against the identified weaknesses. Of course whether they should be actually implemented for a particular application depends on factors such as the reason for investigation, an assessment of the risk associated with the weakness, and considerations of prioritisation and financial cost of the protections.

Retrospective

A simplified form of PERE has been used by Adelard, a safety consultancy, but it has not been further developed or evaluated. We believe that the principles are still sound but the number of organisations that consider the safety aspects of their processes is so small that it is unlikely to be further developed.

Chapter 13

Responsibility Modelling

Russell Lock, Loughborough University; Ian Sommerville, University of St Andrews and Tim Storer, University of Glasgow

Summary

Responsibility Modelling (RM) is a graphical modelling and analysis technique designed to help people record and analyse responsibilities within organisations, to explore the structure and dependability of socio-technical systems. RM uses âĂŸResponsibility' as a unifying concept to explore the relationships between personnel, technical systems and information resources, within a systems' organisational structure. Associated with responsibilities are agents, who discharge the responsibility, and resources, which are used by agents. The graphical notation is accompanied by a risk analysis technique designed to improve dependability and resilience within the socio-technical system.

Background

Our work on responsibility modelling has been based on our socio-technical view of system dependability where we believe that the best way to improve dependability is to consider how people work with computers to achieve dependability. We were looking for a unified way to model socio-technical systems that could relate the human and automated agents in the system to their working environment.

Technical components are ideally suited towards consistent undertaking of repetitive tasks. Human operators, with their greater flexibility, can often adapt to unplanned situations before failures manifest themselves. The notion that human agents in a system, employed appropriately, can contribute positively to the dependability of technical systems is one that is often missed in discussions of software dependability.

For our purposes, we define a responsibility as:

A duty, held by some agent, to achieve, maintain or avoid some given state, subject to conformance with organisational, social and cultural norms.

The term âĂŸduty' in this context refers to both the undertaking of activities, and accountability for those activities. The phrase organisational, social and cultural norms relates to the inherent nature of responsibilities; that systems are adapted to fit the organisational culture they operate in, and that processes are subject to both social and legal compliance. RM was first proposed during the development of the ORDIT methodology in 1993. ORDIT defined a graphical notation to describe the responsibilities held between human agents within socio-technical systems. The ideas were developed in the DIRC project and documented in a book entitled Responsibility and Dependable Systems. Further development has taken place since then at the universities of St Andrews, Loughborough, York and Glasgow.

RM is designed to model responsibilities across complex organisations, which could be real organisations or âĂŸvirtual organisations' that encompass several organisations working together on a shared problem. An example of a virtual organisation is the team that is created to cope with civil emergencies where several emergency services work with local authorities to cope with emergencies such as flooding, terrorist attacks, major accidents, etc. Contingency plans are drawn up in advance of an incident, but such plans are wordy documents that are often inconsistent and incomplete. We have investigated how to use responsibility modelling to represent these plans with a view to making them more accessible (and hence easier to analyse by experts) and to discover potential vulnerabilities that could result in system failures.

By exploring the dependencies between responsibilities and human, technical and information resources, a number of areas can be explored. For example:

1. Who is responsible for updating a given information resource?

2. Who uses that resource?

3. What training people require to access that resource?

Responsibility modelling provides a modelling technique that helps ensure that, for example, the contradictory views of agents, and unallocated responsibilities / resources are identified and discussed. The research at St Andrews extended this by allowing end users to explore the risks associated with deviation from the expected within a given system. By doing so the dependability and resilience of the system can be explored with reference to: Analysing the current configuration of a system to determine what improvements can / should be made on an ongoing, periodic basis.

In the event of evolution or unanticipated change, examining âĂŸbefore' and âĂŸafter' analysis to determine what effect this has had on the ongoing dependability and resilience of a system. Effectively measuring the dependability and resilience of a socio-technical system itself is complex. The distinction between failure and success is unlikely to be clear within a socio-technical system. As such reliability metrics and such as MTTF (Mean Time To Failure), MTTR (Mean Time To Recovery) etc, are of limited use.

In these situations it is more appropriate to apply vulnerability analysis techniques similar to those used in dependability and safety cases to illustrate the strength of the system from the perspective of its processes, training and management. Whilst applicable to both technical and socio-technical systems, dependability / safety Cases require expert construction, an often unreasonable approach outside the safety critical domain, where resources are limited, and such techniques are not mandatory.

HAZOPS (Hazard and Operability Study) is an approach to vulnerability analysis originally developed by ICI in the 1970s, for use in the chemical industry which has been applied to wider domains, including work on socio-technical systems. HAZOPS focuses on the identification of potential vulnerabilities using keywords and associated risks through in-depth descriptions of the system in question, with a focus on technical operability and efficiency. HAZOPS keywords are used to construct tables examining the effect of deviation from the norm for a given process. For example: given a specific deviation for a given process, (something occurring early, late, never, in reverse, too much etc); what are the consequences; what actions could be taken to mitigate the consequences; what safeguards could be put in place; what is the risk of occurrence etc.

RM applies an adapted HAZOPS approach designed to achieve much of the assurance provided by standard HAZOPS whilst being less sensitive to incomplete information, and through the use of more limited generic categories of hazard, which have been tailored towards the concerns of system evolution.

The key components of a responsibility model are:

1. Responsibilities, indicated by a round-edged rectangle.

2. Agents (named in pointy brackets) who are assigned responsibilities and who take actions to âĂŸdischarge' these responsibilities. Agents may be individuals, roles or organisations.

3. Resources which are used in the discharge of a responsibility (named within straight brackets). These may be shared information or may be physical resources such as tools or vehicles.

Figure 1 is an example RM diagram based on an analysis of a flood contingency plan for Carlisle in northern England. Notice that the responsibility 'Collect Evacuee Information' does not have an agent associated with it. Drawing up the responsibility model revealed this vulnerability in the emergency plan, since it did not define which agency should collect this information.

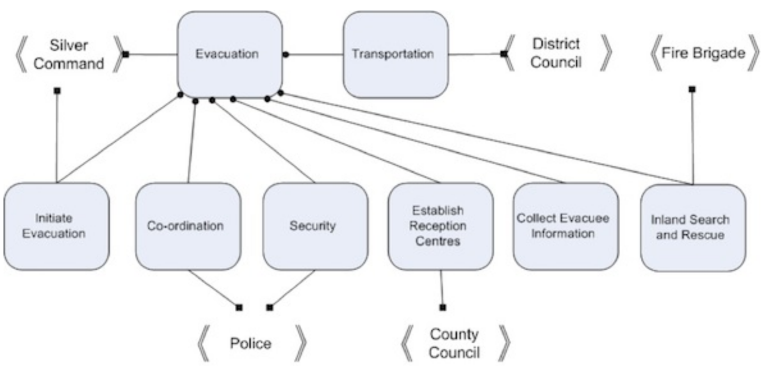


Figure 1: Example RM diagram for flooding evacuation

Resources can also be associated with responsibilities as shown in Figure 2. In this case, the responsibility 'Initiate Evacuation' requires information resources (information about risk assessments and flood warnings) to discharge the responsibility.

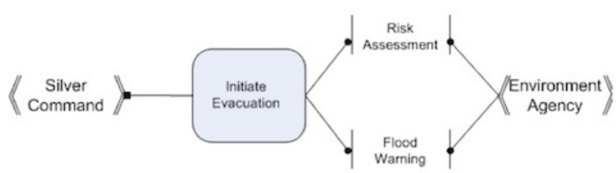


Figure 2: Example RM diagram for resource association

Risk Analysis

Responsibility modelling uses a modified HAZOPS approach to facilitate end user exploration of the vulnerabilities and associated risks of system configurations. Clauses are used, as in HAZOPS, to outline the risks associated with events/potential events. Clauses can then be grouped into tables exploring similar situations. The discursive analysis of clauses can be used to promote:

Risk Avoidance (determining how to eliminate a given hazard) Risk Minimisation (reducing the risk of hazard occurrence) Risk Mitigation (determining how to deal with the consequences of hazard occurrence effectively)

Each clause contains the following information:

Target: The entity to which the clause refers, for example, a responsibility or physical resource.

Context: What is occurring, for example, has there been a flood?

Hazard: A restricted set of hazard keywords designed for generic use within RM models:

Early: The availability of resources before they are required

Late: The availability of resources after they were required

Never: By exploring the effect of permanent failure system resilience can be explored.

Insufficient: Occurrence at an inappropriate rate / level. The types of potential issue are broad and include: Insufficient management; Insufficient maintenance; Insufficient training; Insufficient process capacity; Insufficient physical resources within a given system; Insufficient information flow (in terms of depth and/or frequency)

Incorrect: The effect of incorrect information within SoS can be far reaching and potentially even life threatening. For example, during investigations into contingency flood plans within Cumbria it was discovered that lists used by multiple organisations, of people to be evacuated from flood risk areas in the event of serious flooding, did not take into account transitory residents using caravan parks, effectively designating them locations with no population.

Risk: Risk is defined as a combination of the probability of the hazard and the severity of the hazard occurring. While probabilistic measurement would give the best basis for comparison and analysis, it is likely to be beyond the capabilities of untrained users to either generate consistently or reason about. Instead, qualitative statements are preferable, as categories specific to a given domain can be formulated and applied in a more consistent manner.

Consequences: The potential effects of the hazard manifesting itself in the wider system.

Recommended Actions: The cause(s) of action, either mitigation or avoidance, that could be taken to deal with the situation in question. Whether a given course of action should be taken is tempered by economic, organisational and political factors.

Responsibility model use

RM has been applied to a number of domains ranging from contingency planning to system procurement. There are a number of promising avenues for future research, including those of system simulation and SoS (System of System) modelling. RM is a technique backed up by tools constructed over a number of years to support those developing and discussing models, with the added benefits of semi automated analysis for potential problems. Responsibility models have also been used as a basis for understanding the requirements for the information that is required by agents in discharging a responsibility. A set of standard questions is used to derive some of the requirements for systems that may be used to support agents who have been assigned responsibilities.

The standard questions are:

1. What information is required to discharge this responsibility?

2. What channels are used to communicate this information?

3. Where does this information come from?

4. What information is recorded in the discharge of this responsibility and why?

5. What channels are used to communicate this recorded information?

6. What are the consequences if the information required is unavailable, inaccurate, incomplete, late, early?

The notation used in responsibility models

RM uses a number of key entities and relationships. Figure 3 illustrates the entities and relationships, with further information on each provided below.

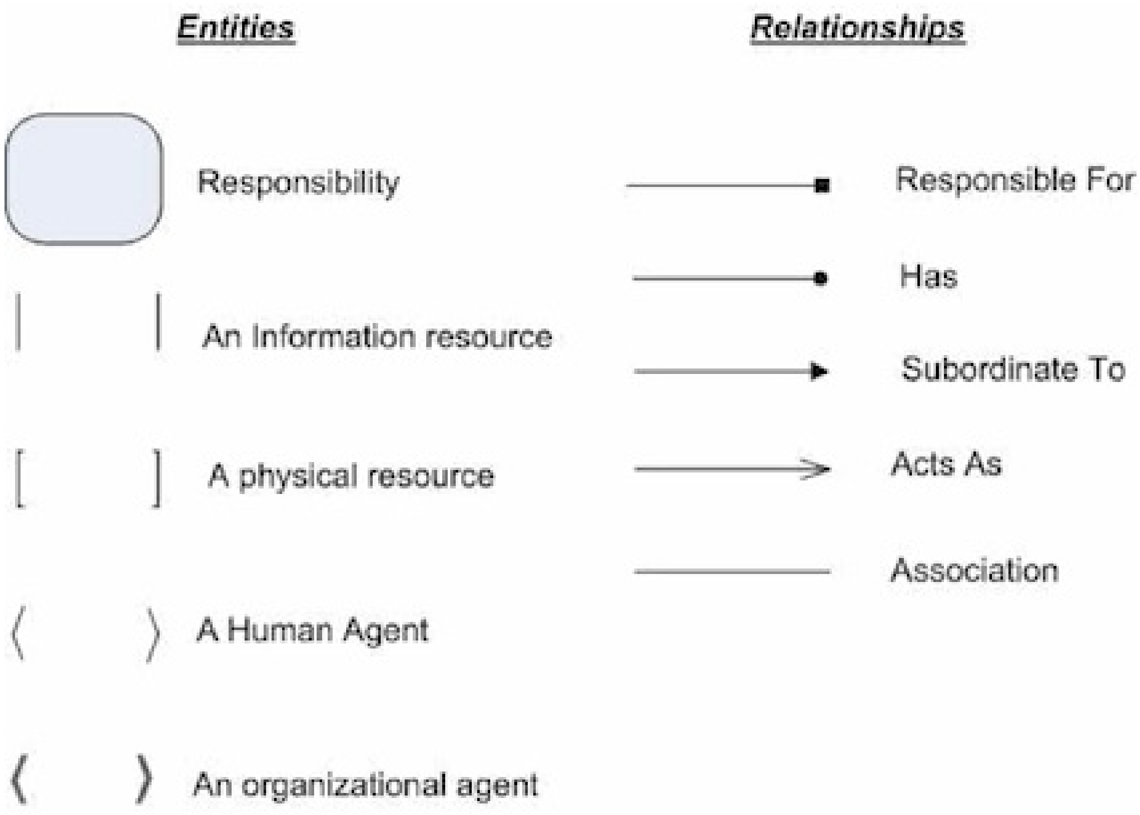


Figure 3: The entity and relationship types within RM

Key **Responsibility**: A stated activity or abstract concept. For example, raising an alarm.

Information Resource: For example, a report or database.

Physical Resource: For example, a piece of equipment such as a PC.

Human Agent: For example, an administrator Organisational Agent: For example, the government, the NHS etc. **Responsible For**: The allocation of an agent to a responsibility.

Has: The allocation of resources to agents or responsibilities.

Subordinate To: To model organisational hierarchies.

Acts As: For example, Bob acts as an administrator.

Association: Used to annotate models with relationships of a domain specific type not covered explicitly by the notation.

Retrospective

Responsibility modelling is a relative recent development and it is still evolving as a practical approach to modelling organisations. Our experiments so far suggest that the notion of responsibility is practical and intuitively understandable and the responsibility modelling can be applied in a range of different domains. So far, the approach has proved to be most useful as a tool for analysing responsibilities and further work is required on how it can be used constructively to help in the organisational change process where new responsibilities are planned.

Chapter 14

Organisations

John Rooksby, University of St Andrews

Summary

Systems engineering for organisations presents many socio-technical challenges. These challenges are not just problems of better specifications and better technologies, but also of understanding, supporting and changing the real world practices of the people working in organisations.

Background

Engineering information technology for organisations is a socio-technical problem. This is because IT is often developed as a part of organisational change projects, because IT impacts the ways people work and interact, and because IT must be adequately supported within an organisation. Therefore, systems engineers need not only understand and configure technology, but must simultaneously be able to understand and configure organisations.

The word organisation can be used as both a noun and a verb. Socio-technical systems research views organisations in both of these senses, both as a thing and an activity. Organisations are not static, objective entities, but are enacted, constantly being held together by the actions, talk, writing, decisions, and so on, of those working in and around them. Consider a University; you could point at various departments, lecture theatres, sports facilities, libraries and so on, but what makes it a University is not just its buildings and equipment but also the work of those within it (such as staff and students, etc.), and those around it (such as Government departments, funding agencies, etc.). For socio-technical systems engineering, it is important to see organisations in both of these senses; systems must fit with and support both what the organisation is, and what the organisation does.

Organisational Models

An organisation will usually own models and descriptions of its organisational structure and of the roles, responsibilities and processes that take place within it. However the realities of an organisation will hold differences to those representations. The reasons for this include:

- Organisations are complex and difficult to represent.
- Organisations constantly change and so the representations quickly date.
- The ways in which people work are contingent upon contextual factors such as the time and resources available, their training, and whether there are other competing demands.
- Much of the work of an organisation is (rightly or wrongly) seen as too mundane or unremarkable to model.
- Representations of organisations are often aspirational, signalling what the organisation is seeking to become.

Therefore it should never be assumed that a model of an organisation is accurate. This is not however to say that such representations should be abandoned, but that they should be seen as indicative rather than descriptive. Problems can be caused when information technologies are implemented that support the organisation-as-modelled but not the organisation-as-practiced. Socio-technical systems engineering needs to account for both of these.

Work Practices in Organisations

Organisations often employ many people, and they are often distributed across different sites. A routine challenge faced by organisations therefore is how work can be kept orderly between employees; how people can collaborate, work cooperatively around shared resources, minimise conflict and overlap, etc. This is handled by a combination of formal contrivances and more informal, social practices. For example:

- Organisational work often involves following plans. These allow for work to be scheduled, articulated and tracked. However these support, but do not ensure orderliness. For example a plan cannot provide remedies for all contingencies. Plans are routinely followed dynamically and remade as work progresses. Questions can repeatedly arise as to what exactly can be done to satisfy the plan, what parts of the plan are achievable given the time available, and what is missing from the plan.

- Some organisational work follows defined processes or workflows. For example call centre workers follow scripts, health professionals follow clinical protocols and workgroups may be coordinated through workflow systems. To follow these, it is often necessary to do work that is not defined by the process itself, for example extra work or checks may be needed between steps and people may need to backtrack or look ahead in a process. To diverge from or abandon these is also often done for good reasons, perhaps because they do not fit with the current problem being addressed, or because the necessary resources are not available.

- Although projects, plans, procedures and so on may be presented as abstractions, as manuals, as flowcharts, and so on, just what it takes to realise them is a practical matter that must address the contingencies that will inevitably arise. Socio-technical systems engineering must therefore pay attention not only to the formal aspects of the division of labour, but the practical ways in which this is handled.

Knowledge and Information in Organisations

The development of information technology for organisations usually has to somehow rationalise the information held and used by that organisation. This can often prove a complex task and will inevitably have partial coverage. The reasons for this include:

- Organisations routinely deal with myriad forms of information, from formal documents to informal notes, from statements of fact to opinions and comments, from processes and workflows to know-how about how something was done before, and so on. Information is made and remade at different times and in different places across the organisation. Documents are often re-written, copied between, marked-up, collected together and so on, and so information becomes held in different versions for different purposes in different places.

- Information is often made meaningful by its context, for example by it's positioning on a desk or in a folder. The meaning of any document or other form of information is not necessarily made clear by that document itself but through its featuring in routine work, through talk and activities around it and through its bundling with other documents.

- Computerisation also often removes the physical affordances of documents, meaning for example that they cannot be handed around, collected together, talked across, and so on. There is rarely one single information system used, but information will be spread across multiple technologies, many of which will overlap and conflict with each other and many of which will be short lived. Attempts to implement a single system for an organisation will often see other technologies developed around it, for example people often use spreadsheets and word processors alongside enterprise systems.

- Information technology is extremely valuable for organisations, but it is not the case that all of the knowledge within an organisation can be formalised. Attempts to formalise information can lead to frustrations and often informal, local stores of information emerge (such as notepads, collections of print-outs, cheat-sheets etc.) to support the work. Technology support for knowledge work therefore needs to address the needs and practices of workers, not just concentrate on the logics of information.

New Technology and Organisational Transformation

Although new technologies are routinely touted as transformational, the reality is that organisational change is not determined by technology. New technologies in organisations go through a process of domestication, in which the people that work with them develop ways of using the technology that they find convenient and suitable for what they must achieve in their work. In this there is often a strong push towards making new technologies into tools to achieve "business as usual". For example, the introduction of new technology in financial services, whether telephone banking, video-links or Internet banking, especially in the context of customer-facing work, has not re-written the relationship between the bank and its customers but necessitated the development of new routines and competencies in customer interaction.

Working Across Organisations

Large organisations or enterprises are usually made up of multiple sub-organisations. Organisations also rarely work in isolation from other external organisations, for example they may work in partnerships, sub-contract, interact with regulatory bodies, have close relationships with suppliers etc. Organisations therefore are rarely discrete entities but overlap and have fuzzy boundaries. Information technology can cause problems here in that it tends to congeal the boundaries around an organisation; in the process of making information available within and unavailable without an organisation, IT entails that "within" and "without" become much more tightly defined.

Retrospective

Organisational issues are often problems of the kind that can be tamed, but not solved. Ethnographic work has served to highlight these problems, and ethnography as an approach within socio-technical systems engineering remains useful as it can address the unique ways in which organisational problems emerge and must be tamed during the systems engineering process. The papers on this topic therefore do not offer solutions but serve to sensitise their readers to the kinds of organisational issues that manifest.

Chapter 15

Health and Social Care

Summary

Socio-technical approaches are often applied in the development of healthcare information systems and medical devices. Socio-technical approaches are appropriate because the problems of developing technology for healthcare lie not with the complexity or novelty of the technology itself, but in the complex ways healthcare is practised and organised.

Background

The healthcare domain has proved unexpectedly complex for systems developers. The production of integrated electronic health records was originally seen as straightforward, no more complex than the production of any records system for any organisation. Only after decades of effort and phenomenal cost are we beginning to see these deployed.

There is also a long list of failures in this domain. A particularly notable failure is that of the London Ambulance dispatch system in 1992. This failure was implicated in the deaths of up to 30 people. This was not a technical, but a socio-technical failure; there were technical problems with this system, but it also failed to address and support the established practices of dispatch. It became clear from this disaster that simply procuring the cheapest system was not an appropriate strategy. Development approaches need to be both technically sound and to pay appropriate attention to the support and evolution of work practices.

The Domain

Healthcare and social care are large, overlapping domains encompassing a wide range of organisations and activities. Needless to say, the work in these domains is often life-critical. Health and social care organisations include:

- **Hospitals**: These are usually large, complex organisations. Hospitals within any health system may do similar tasks but often have vastly different cultures, practices and outcomes.
- **General practices/Family Doctors**: These are often small, autonomous practices. They are usually a first point of call for any non-emergency situation, and build long-term relationships with patients.
- **Ambulance Services**: These are not usually operated from a hospital but by an independent organisation.
- **Pharmacies**: These are independent businesses whose work it is to prepare and dispense medications.
- **Social care**: This is a large area, encompassing mental health services, patient rehabilitation, care for elders and so on. Failures of social care can lead to problems that need to be addressed by other services.
- **Public health**: Many governments and charity organisations run public health initiatives with the aim of preventing people from developing health problems in the first place.

Healthcare organisations can vastly improve the outcomes of care when they communicate and coordinate effectively. When someone falls ill, for example, they may interact with several organisations, and be discharged from hospital into social care. So information technology problems do not just fall into each of the areas mentioned above, but they also exist across them.

Complexities in Health and Social Care

Why is developing information technology for health and social care so difficult? Why is this sector such a quagmire for technology with project after project going over budget and failing to deliver what was envisaged? The answers to such questions seem to lie in the complexity of the domain:

1. There is simply a mass of information. Everyone in the developed world will engage with a healthcare provider many times during their lifetime. For each engagement, relevant information from previous ones should be available. Each encounter is also potentially useful in building evidence about the efficacy of services and treatments.

2. Healthcare organisations are extremely large. Organisations such as NHS England, NHS Scotland, and Kaiser Permanente in the USA operate many hospitals and services. Hospitals are themselves large organisations, and often act in a highly autonomous way.

3. Healthcare is highly politicised, with successive governments trying to reform and reshape the sector.

4. Powerful professional groups dominate healthcare. Many of these groups predate the organisations they work within. These groups cannot be managed in the same way as employees of other organisations often can be, health professionals must be persuaded rather than told what to do.

5. Healthcare faces a massive coordination problem, this ranges from co-operation in operating theatres, to shift handovers, to shared care and handover between organisations. The quality of coordination can have a huge impact on the effectiveness of care, for example a fast coordinated response to a stroke and then coordinated, long-term rehabilitation massively increases the chance of a stroke victim fully recovering.

6. Finally, the range and complexity of tasks undertaken in healthcare is huge. These tasks are also highly contingent, particularly where health professionals need to deal with acute situations.

Socio-technical Challenges for Health and Social Care

Here we list four major socio-technical challenges for systems engineering in health and social care.

1. Supporting Practice

It is extremely important that information technology for healthcare is fit for purpose. Hospitals in particular are complex work environments, where health professionals work with multiple technologies, paper, devices, with other people, and not least patients. The preference in systems development is to design and evaluate a technology under controlled conditions, but this ignores the complexity and dynamic nature of the work and environment. Figure 1 below shows just how many technologies may be in use at any one time. Somewhere beneath all the equipment is a premature baby fighting for survival.

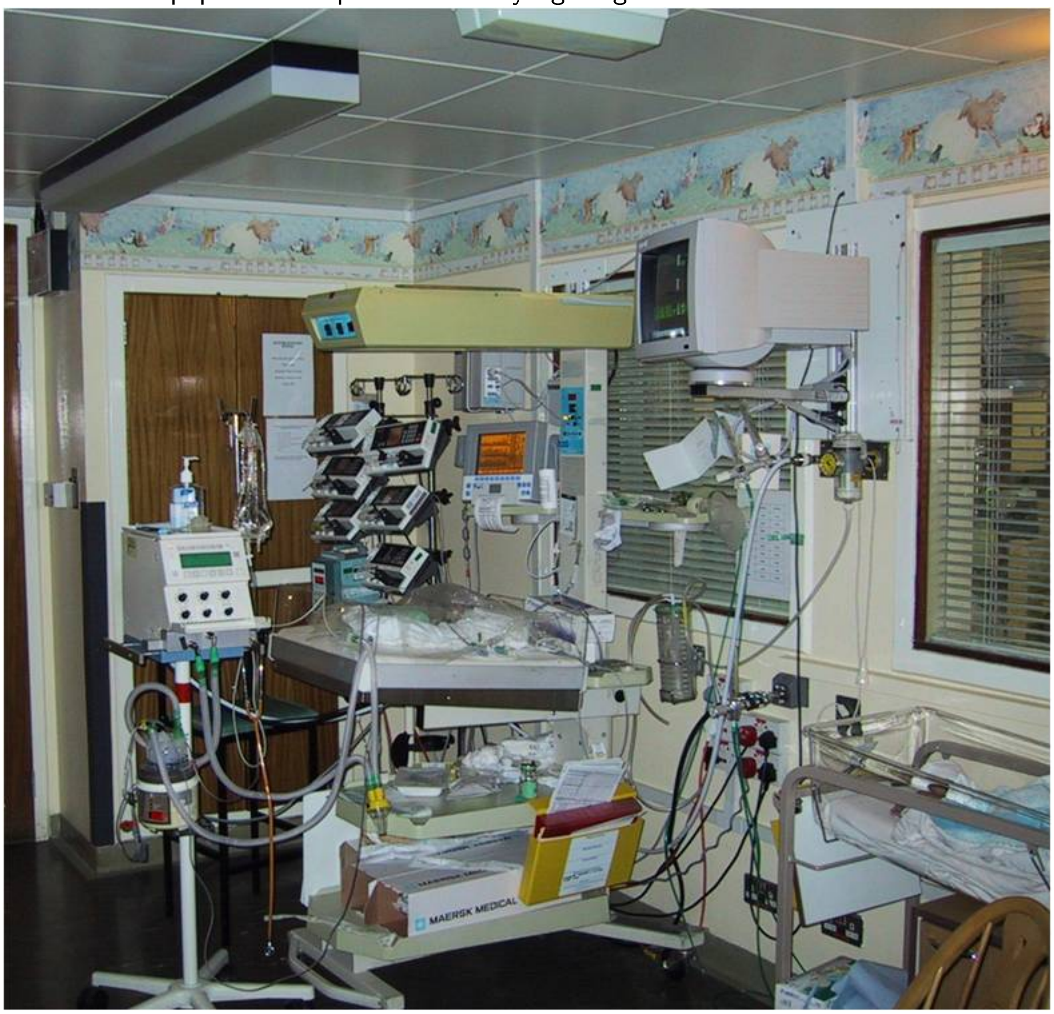


Figure 1: A Neo-natal Intensive Care Unit

2. The Paperless Hospital

One of the key challenges in healthcare is to produce an integrated electronic record. This has been a long-term aspiration, proving surprisingly difficult. Only recently have electronic records been coming into widespread use, and still there are many problems in ensuring the correct people have access to these, that they are used in uniform ways, and that the information in them is correct and meaningful. Summary records to be shared between providers are still "just around the corner"(as they have been for a long time). One of the key problems of computerisation has been that a medical record is not a thing, but a collection. Medical records are collections of results, summaries, letters and so on. Another problem has been that paper is very convenient for care providers to use. Health professionals do not sit at computers all day but face patients and work with others. Having a piece of paper on a clipboard is extremely convenient. Thirdly, the record has historically been, not simply information about a patient, but notes written to be used among small, richly connected groups. Figure 2 is used to illustrate this point.

FINDINGS: There are aneurysms arising from both intracavernous internal carotid arteries. On the left, the immediate pre-cavernous and intracavernous portions of the internal carotid artery are dysplastic with a fusiform aneurysm. This has a maximum dimension of approximately 1cm. On the right, there is a larger more saccular aneurysm with a maximum dimension of 2cm. This also arises from a dysplastic intracavernous internal carotid artery. The source data images from the angiography and axial GRASE images demonstrate these aneurysms nicely lying within the cavernous sinuses. In addition, there are changes of small vessel cerebrovascular disease in the brain with small lacunar infarcts involving the right gangliocapsular region.

Figure 2: A Neuroradiology Report

Figure 2 shows a section from a neuroradiology report. It holds a great deal of technical language, but is not simply a âĂŸcontainer of facts'. Consider the part of the findings section that states "the angiography and axial GRASE images demonstrate these aneurysms nicely lying within the cavernous sinuses". Why would an aneurysm ever be described as "nicely lying" anywhere in someone's body? The answer, of course, is that this report is oriented to the medical procedures that are to follow. Despite its formal nature, this report is more like a letter written to a person the author knows well, and is relevant to what the author thinks that person needs to know. Computer scientists often approach medial information as if it is a series of facts, but in reality it is highly bound up among people and practices.

3. Large Scale Information Systems

The scale of healthcare means that the information systems that support it can be very large scale. In practice there are many independently produced and managed systems that are expected to be interoperable.

In England, the National Programme for IT (NPfIT) was established in 2002 to see the development of a number of technologies for the NHS including integrated electronic records, electronic prescribing and networking infrastructure. It was originally intended as a three-year project costing 2.3 billion pounds. However this spiralled to 12.4 billion over 10 years. Some elements of the programme have been very successful (if over time and over budget). Other health services, for example the NHS in Scotland, have sought to undertake more bottom-up driven programmes, with technologies being developed at a local level and the being scaled up to national level if they are successful. Both bottom-up and top-down strategies run into problems, and it is not clear which is really the better approach. Large programmes such as NPfIT can serve to highlight the cost and difficulty of producing health technologies, but this does not mean other approaches aren't wasteful. Denmark has been widely credited as having the most effective healthcare information technology in the world. In Denmark, there has been a centralised programme of IT governance, but no fixed initiatives. Digitization has been incremental and technologies allowed to evolve. Denmark is of a similar size to Scotland, but it is not clear whether its approaches to healthcare can scale to the larger health systems in England, the USA, etc.

In addition to clinical information, health organisations also produce and manage a great deal of service data. This includes information about demographics, outcomes, medications, care pathways, incidents, episodes of care, and so on. These are large-scale non-trivial data sets that are essential to evaluating, planning and costing services. Handling and making sense of this "big data" is one of the coming major challenges for systems engineering. It is a socio-technical challenge because attention needs to be paid to the ways it is collected, and inevitable problems and inconsistencies in the quality of the data mean that support needs to be provided for how it can be legitimately interpreted.

4. Patient Safety

Another key challenge of developing medical technologies is to improve and support patient safety. Medical error is a leading cause of death and injury. In the USA, more people die every year from medical error than they do road accidents. In England up to 5% of hospital admissions suffer a preventable error and 0.3% of admissions die from these. Such preventable errors are unacceptable, and the cost of litigation against healthcare providers is actually far higher than the costs of implementing safety precautions. Information technology cannot be used to solve all types of preventable accident (for example patient falls, or hospital acquired infections) but it is relevant to many diagnostic and treatment errors.

A major effort has been made to address medication safety. Medication error is a major source of harm, and a great deal of attention has been paid here to how information can be clearly recorded and presented, checks made and problems tracked. The switch to computerised order entry has been one of the most effective means of reducing medication error. Computerised entry resolves problems with the legibility of handwriting, missing fields, inappropriate values entered and so on. A more complicated challenge is support for flagging potential adverse drug interactions where more than one medication is prescribed for a patient. Another challenge is the linking of prescribing systems to patient records.

The large-scale collection and examination of incident reports has also been a key way in which information technology can be used to improve safety. Incident reporting was brought to healthcare after its success in the aviation industry. Anaesthesiology was the first profession to adopt this on a broad scale, but now incident reporting systems are usually organisation wide. Incident reports are used to report incidents of harm or where there was potential for harm. Reporting these enables health providers to learn from problems and mistakes and put corrective measures in place. It is important to learn about small problems, and incidents where there was no actual harm. Major accidents often share causes with smaller incidents. Major accidents are rarely completely novel events, but result from unfortunate combinations of factors that have previously caused no harm.

Retrospective

Socio-technical work has repeatedly stressed the importance of understanding local, practical issues in the design, configuration and deployment of healthcare technologies. Technologies continue to fail because they do not suit the complexities of practice and/or because they are unacceptable to those who must use them. This has never been to deny that generic technologies are inappropriate for care, but is to say deployment is dependent upon local factors.

Over the next decade, socio-technical systems engineering for health and social care will increasingly face challenges associated with large-scale systems engineering. How can coordination be supported across systems and between organisations? How can large-scale networks and datasets be managed? And how can internet based services be better used by the public for gathering information and storing health related data (for example from digital devices)?

Chapter 16

Control Rooms

David Martin, Xerox Research Centre, Grenoble

Summary

Lancaster was one of the pioneering centres for conducting social analysis of different work settings. From the end of the 1980s and onwards the rise of social analysis went alongside the development of networked systems to support cooperative work. Social analysis, particularly ethnographic (observational) fieldwork, is considered a particularly well suited approach to studying cooperative and social aspects of work, to be supported by technology. Furthermore, there was a growing interest in multi-media sites and applications. Due to their variety of technologies and for their concentration of staff within a circumscribed area, control rooms were very popular for conducting studies âĂŞ indeed, they were known as the multi-media field sites par excellence. Attention was also drawn to these sites for their safety and time critical nature âĂŞ mistakes could have a high impact so dependable technologies and dependable working practices were crucial.

Background

Control room studies were hallmark studies in the discipline of Computer Supported Cooperative Work (CSCW), a discipline that grew out of the 1980's as networked computers and shared applications offered new possibilities for people to work together, at a distance. One relatively novel aspect of CSCW was that with its focus on cooperative and social use of computers, sociologists began to work within the discipline, pointing out that that if you want to design systems to support social and cooperative aspects of work, you might want to first have a good understanding of how the mechanics of these play out in current work settings, potentially the ones that you were looking to augment with new technologies, distribute and so on. In a confluence of good fortune and planning in the late 1980s a number of field study opportunities arose within projects focusing on control rooms.

At Lancaster University Dave Randall and Richard Harper conducted field studies of air traffic control room, working alongside other sociologists and computer scientists, led by John Hughes and Ian Sommerville, working in a truly mixed team. They produced a series of articles into the 1990s. At Kings College London a similar study was being conducted of London Underground control by Christian Heath and Paul Luff. Over the following years more studies were carried out in air traffic control by Lucy Suchman, Wendy Mackay and Maria Normark while studies also focused on other control rooms, notably a series of studies on ambulance control conducted in Manchester, UK (Martin) and Sweden, (Pettersen, Normark). The series of control room studies have proved very fruitful being as they are particularly suitable for conducting ethnographic studies as most things are happening in the one place and they are rich in social interaction and in technology âĂŞ but even more so they have provided some of the key empirical findings within the discipline of CSCW. And these findings have had a strong influence on elements of technical development over the years.

Air Traffic Control and London Underground Control

The control room studies focused on the details of cooperation and coordination of workers in situations that were often time critical. A second key feature of the studies was the examination of the roles of different technologies, some basic some more high-tech. Of particular interest was the ways in which workers interacted with technologies or worked with artefacts in a manner that cooperative work was achieved through the artefact with minimal need for control room workers to engage in overt conversations about the activity. Years of working together had allowed the workers to rather seamlessly coordinate their actions through observing the placement of artefacts and people's interaction with them.

To provide two concrete examples: firstly, in air traffic control the researchers drew attention to the use of flight strips. These paper strips contained all the key information regarding a plan and flight âĂŞ flight number, carrier, times, flight path etc. As information about the flight changed âĂŞ i.e. how it was being directed by the controllers, timings etc. âĂŞ so the strip was amended, and in this way with its alterations the workers could see how the flight had changed and who had made the decisions. Another feature of the flight strips were that they were displayed on a wall, meaning workers could see the list of flights to deal with at-a-glance, and problem strips were even âĂŸcocked-out' (made visible) so workers in general could orient to the issue. These features of the strips allowed people to see the work undertaken by others, their decisions and to orient to issues as a group.

The second example comes from London Underground Control. In this situation you have a controller (in charge of the management of the train schedule) and an announcer (who lets passengers waiting at stations know when to expect the next train) working together in the control room. The main technologies used in this setting for the controller were audio and video links, the âĂŸfixed line diagram' (a strip of lights that showed the real-time progress of trains towards the station) and a paper timetable of planned train times.

A significant feature of the work is that the timetable serves like a basic plan but that due to all the contingencies of the Underground system the workers need to modify and update the schedule on-the-fly due to small delays, problems, incidents and so forth. This often means updates whereby trains swap places, are late etc. This study focused on the way in which the controller and announcer could coordinate their work in a fine-grained manner, such that the announcer could enchain his tasks of letting the public know almost directly on the actions of the announcer. This was achieved through both having a shared view on the fixed-line diagram but also through their proximity, the announcer could listen in to the conversations of the controller with drivers and prepare and make his announcements accordingly. The controller could also make elements of his work more visible, by formulating his words in conversations to be overheard, by looking pointedly at the fixed-line diagram and even stressing or emphasising actions. In these ways close coordination was achieved.

Key Findings and Ideas Arising

In these control room studies, therefore, some key concepts were beginning to be developed. These focused on the means by which workers monitored visually or listened into one another's work as a means of understanding what was going on and planning further actions but also reciprocally how workers made available to others features of their on-going work such that others could pick up on actions and act themselves. Other key feature of this work was the role of technology in supporting these shared understandings âĂŞ sometimes, just from watching interaction, sometimes the modification or display of even rudimentary technologies like flight strips. These phenomena became subsumed under the concept of âĂŸawareness' within CSCW. People became very concerned that this was a key concept to support when working at a distance on shared systems, whether synchronously or asynchronously. People cannot pick up on what is going on simply through being collocated in the same environment âĂŞ one needs the system to communicate elements to do with people's activity, what work they have carried out (if on a shared document etc.), maintaining details of what changes have been made by whom, status, even features of their local environment in order for them to work together more fruitfully.

A second key concept that arose from these studies was the coordination or articulation work that actors engaged in, in order to mesh together their tasks. Researchers pointed out that this work to coordinate separate activities was something not often paid attention to in the design of systems. For example, air traffic controllers needed to ensure that they managed the boundaries between different flight sectors as planes travelled between them. And we also have already discussed the importance of the tight integration work between the two separate activities undertaken between the controller and announcer in the London Underground control room. Building on this Martin and colleagues noted how different ambulance dispatchers also worked around the boundaries of their sectors to ensure that ambulance dispatch decisions were appropriate for ensuring rapid response to incidents while maintaining good coverage for the whole of the region. Indeed in times where the service was stretched dispatchers needed to work more as a team, and often their supervisors were also engaged, the group as a whole working together to share information and local knowledge on features such as the drivers levels of stress, road conditions and so on meaning that a good overall service could be maintained.

Coming out of the research on articulation work and awareness was also the finding that having shared artefacts or technologies that allowed actors to see an overall picture of a service status or multiple different views of a service was important to understand what was going on in the service as a whole and what other people they needed to coordinate with were doing. In air traffic control there were the flight strips as well as representations of the air space in sectors. In London Underground control there was the fixed-line diagram. When this was augmented with aural access to on-going conversations the announcer was able to understand what was happening and construct his announcements accordingly. In ambulance control dispatchers could view shared lists of ambulances on call or ready to dispatch for the whole region, as well as a quasi topographic representation of the region showing the status of all ambulances. Working with these together with their own local representations and lists for their area they were able to manage individual and group work. For example, a dispatcher who was not currently very busy could look at her neighbour's workload and offer assistance in the task. This work certainly influenced work concerning the design of what are now better known as shared information spaces âĂŞ technologies for groups to orient to and organise their work.

Comparative Opportunities

As should be clear from the above sections building up a corpus of control room studies was not only useful for the results in themselves and the concepts that became more settled design concerns in CSCW âĂŞ they actually offered great opportunities to synthesise results. Specifically within the domain of air traffic control there was some work looking at how different control rooms functioned, whereby they could be compared and contrasted in terms of their practices and deployment of technology. Furthermore, some articles by Dave Randall and colleagues focused on Swedish ambulance control but offered up a means by which various ambulance studies could be synthesised in terms of the dimensions of âĂŸambiguities', âĂŸawareness' and âĂŸeconomy'.

These were shown to be relevant features of all the studies discussed but played out in different ways responding to sometimes slightly different problems. Field studies have been questioned for their generalising possibilities but across the control room studies a reasonable corpus of knowledge has been built up that means that we often seen similar types of work organisation and technological solutions. This background can be thought of generative in an analytic fashion âĂŞ it aids in the analysis of new settings in a comparative fashion rather than predicting what will be found or prescribing ready-made technological solutions, but nonetheless useful in looking at new settings.

Faltering From Ethnography to Design

We have already pointed to some of the more generic design concepts that have emerged from the control room studies. What was also important about these is that they were some of the first field studies to attempt to use the findings specifically to inform design. Here it is appropriate to single out the work at Lancaster that was published as âĂŸFaltering from Ethnography to Design'. This catalogued a specific attempt by computer scientists to work with and from the ethnographic record to produce new prototype solutions for air traffic control what would be digital technologies that would maintain the functionalities that would support many of the important features of the work. This did not result in the design and deployment of a new air traffic control system but was crucial in demonstrating the possibilities of sociologists and computer scientists working together in a fruitful manner, as well as the possibilities for using field study findings for design.

Retrospective: Moving Out of the Control Room?

As they moved onto new projects the Lancaster group produced a paper titled âĂŸMoving out of the Control Room' in which they outlined four different ways in which field studies could be deployed in settings that were not control rooms, for example to assess the suitability of a proposed technology or to look at features in a complex distributed setting. In many ways this was less a plea that this was what people should be doing than a response that ethnography could handle settings that were not as conducive in terms of ecology and action. And in the following years field studies have been deployed successfully in many different settings. However, it seems fair to say that the control room studies were particularly successful âĂŞ they had a great influence on the young field of CSCW, furnishing it with a number of key concepts and influencing some core technology ideas. And even now people are still going back into the control rooms for their studies today

Chapter 17

Domestic Systems

Ian Sommerville & Guy Dewsbury

Introduction

We now live in a world where most of us have several computer-based systems at home. Home systems are different in many ways from organisational systems - they may be consumer products, it is important that they fit into the fabric of the home and they have to be installed and supported by the users themselves. For older or disabled people, specialised home systems may be 'assistive technology', which supports them in some aspects of their daily lives. Our work, developed as part of the DIRC project has mostly focused on these assistive technologies in the home so that is the theme of this chapter.

The home has different meanings and imbues different feelings to people who live in them. The home provides shelter as well as a sense of identity. Within the home, the domestic space, the way in which participants or occupants interact is very different. The ways in which people use their domestic space is a personal decision, which could be mediated by cultural fashions and other facets of society. For a designer of systems to support people within their domestic space it is important to understand the richness and individuality of the domestic spaces and the meanings that they have to the occupants.

Network connectivity in the home is, in Western societies, becoming universally available. A home network can allow isolated individuals to retain contact with the world around them as well as allow for external monitoring of people who require this service. Social networks, email, newsgroups etc. all allow people to be informed and connected with friends and colleagues. This has important benefits for older and disabled people who may suffer from mobility and other problems that makes it harder for them to meet people face to face.

Dependable assistive technology

Assistive technology is the name given to any device or system that supports everyday living. Generally, they are used by people who have some kind of functional or cognitive impairment - they may be hard of hearing, forgetful, have mobility problems, etc. We are interested in assistive technologies that are computer-based and networked, rather than, e.g. mobility support devices or hearing aids.

Because such devices may be essential to support normal living, it is important that they are dependable, as failure or unavailability can cause harm to their users. In the home of a disabled person, the failure of assistive technology devices could be critical to the occupants. If a piece of telecare, such as a fall detector, fails the person could be lying on the floor for a long time until they are discovered by which time the possibility of hyperthermia and the likelihood of successfully repairing any damaged limbs are

But dependability in this context has to be considered in a rather different way from the dependability of organisational systems. We have a body of knowledge on dependability that can be applied in designing organisational systems but this needs to be extended for domestic systems. Key differences that have to be taken into account include:

1. Activities and processes are standardised in organisational systems but not in home systems. The most significant difference between the organisational system and the home system is that processes and timing functions are different. Home routines are often unplanned and unpredictable.

2. Within the home there are fewer external legislative and advisory bodies imposing their regulations, opinions or choices upon the occupiers.

3. When a device fails in an organisational system it can be very costly as the organisation could consequently lose a considerable level of productivity. In the home, by contrast, if a device fails there is usually a work around.

4. Traditional hazard analysis for organisational systems considers safety and risk from concrete positions where organisations could fall foul of legal problems by not following standards. Risk tends to be minimised at all levels and personnel are prohibited from undertaking various tasks on the grounds of health and safety. Safety and risk are considered part of the organisational framework and their minimisation is central to effective

management. In the home, users may be willing to trade off risk against convenience.

In the home, there are trade-offs between privacy and safety that exist so that users can maintain an acceptable quality of life. Although most people might not wish to be 'spied upon', when their health or even life is in danger, being spied upon might be preferable to being left completely alone with no assistance available. The most appropriate system should always balance privacy with safety.

Finally, system dependability as well as system reliability and system availability are characterised differently from more formal organisational systems such as offices and other businesses. The notion of failure may be less clearly defined and, as systems are assistive, users may be able to cope with unavailability.

Analysis

The home presents a range of potential exploratory areas, which can be themed together to produce activity centres, which result from chains of actions. Activity centres and the chains of actions can be used to articulate the organisation of coordination, how routines and practices are interpreted by individuals within their living spaces. The patterns of activity within a domestic environment are constantly changing as needs and wishes of individuals change. These patterns and rhythms of daily activity are central to design, particularly for designs for disabled people.

The variation of users and personalisation of activities within the home mean that, for assistive technologies, analysis of each setting is important before technologies are chosen, designed and installed. Conventional analysis for organisational systems, based on a technology or process perspective is inappropriate and it is essential that a socio-technical approach to analysis is adopted.

However, socio-technical analysis designed for organisational systems such as workplace ethnographies often require considerable on-site working by an observer and this may be impractical in the home. Furthermore, interviews with participants which reply on their memories are likely to be unreliable, with essential details forgotten or considered by the users to be unimportant.

For this reason, we adapted the notion of cultural probes, first developed to facilitate the design of systems for elderly people, to support the analysis of domestic system requirements for elderly people. We provided users with a kit of different ways to record their everyday actions - a camera, a notebook, a diary, a sound recorder and asked them to use these over a period of time. We then conducted interviews with the users which were focused on the records that they had made. This meant that we had information about their everyday lives without the need for intrusive analysis.

A dependability model for domestic systems

The potential criticality of home systems for disabled people is of great concern. Any design, which has a person or people relying on the system for support and quality of life becomes a critical system. Therefore, the design and use of assistive technology in the home should be viewed from the perspective of the assistive technology as a critical intervention. To help guide the design process and to bring dependability issues to the forefront of the designer's mind, we developed a dependability model for domestic systems as shown in Figure 1.

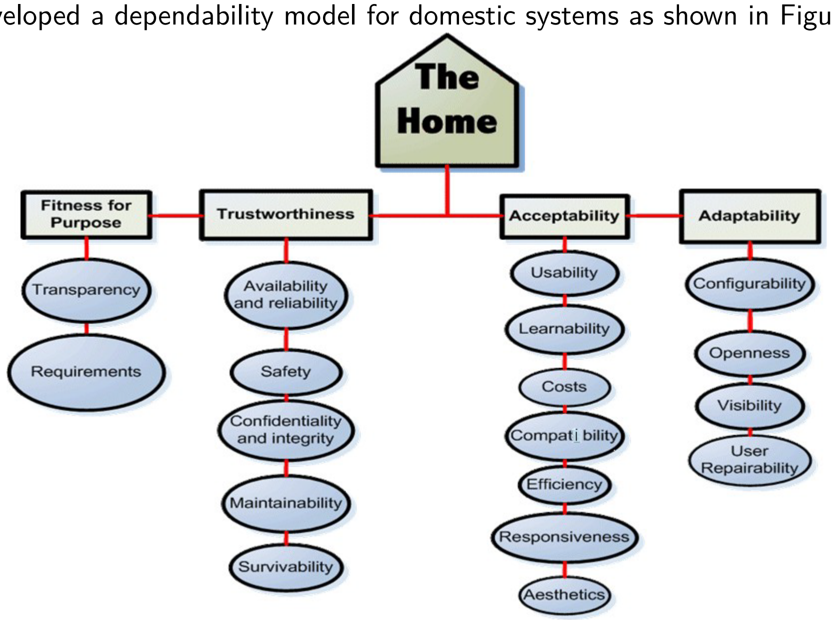


Figure 1. A dependability model for domestic systems

The Dependability Model of Domestic System (DMDS) was developed to illustrate the key areas of importance in the design of assistive technology systems. This in turn led to the development of MDDS (Method of Dependable Domestic Systems) which features a set of tools to facilitate system dependability for assistive technology systems. MDDS became simplified further to produce DTA (The Dependability Telecare Assessment Tool) which was developed to facilitate dependability in telecare systems.

The attributes in the domestic dependability model reflect system attributes that may be important in a domestic setting. They are grouped under 4 headings:

1. **Fitness for purpose**. The system must meet the needs of the user, which is not the same, of course, as meeting the requirements for the system as defined to the designer. A device chosen may be the wrong device for that user, even although it performs according to its specification.

2. **Trustworthiness**. The system must be trusted by its users otherwise they will simply turn it off - something that organisational users of systems may not be able to do.

3. **Acceptability**. The system has to fit in with the ways in which users live their lives and what is important to them. It is not just a question of 'does the device work' but also whether or not users are willing to change their homes and the way they live to accommodate the device.

4. **Adaptability**. The system has to be able to evolve to reflect the changing needs to users. For older and disabled users, their capabilities change over time and the device has to be able to evolve to reflect this.

We have covered these issues in depth on our paper on a dependability model for domestic systems (link here to paper) and space does not allow us to describe all of them in detail here. However, to help explain why dependability for home systems is different, consider two of what we consider to be the most important attributes - user repairability and aesthetics.

User repairability is an essential quality of devices in the home, whereas organisational users call on technicians to repair their systems. Repairability is distinct from maintainability, which reflects the ability of the system to evolve to reflect changing user needs. Domestic consumer devices are usually built to be available for a relatively low price and this inevitably means that the components of these devices have to be low cost. Breakdowns are therefore likely to be more common than in more expensive organisational systems. To maintain dependability (and to ensure that users do not incur unacceptable maintenance costs), users should be able to repair devices themselves - at least to some extent. This requirement leads to design considerations such as:

1. All devices should have their self-diagnostic functions built into them.
2. To recover from user errors, all devices should include a simple mechanism to reset and restart the system.
3. Devices should be accessible - it should be possible to electrically isolate devices, safely remove covers and access components without specialist tools.

You may think that aesthetics is an unusual and unnecessary system dependability attribute. But, for many people, the way their home looks and the way that assistive technology devices fit into their home is important. Devices are usually purchased to complement the domestic physical environment, and the self-image of the user. If aesthetically unpleasing devices are proposed or installed, the user may simply refuse to accept or to use them. They are therefore unavailable and so the overall dependability of the system is compromised.

More information on domestic system dependability is available from this blog: http://thetelecareblog.blogspot.com/

Chapter 18

Banking and Finance

The importance of the social in socio-technical systems and some of the fundamental insights concerning the importance of the social in socio-technical systems, can be evidently seen in studies of financial organizations and their use of technology.

Banks, and financial services more generally, – since few banks merely store, dispense and lend money but instead have expanded their operations to include a range of financial services such as mortgages, share dealing, insurance of various kinds - have increasingly been seen to be at the sharp end of global economic transformations. These obviously include the recent financial crisis, the enforced 'bail-out' of the banks and the ongoing economic crisis but also include other, less dramatic precursors - processes of 'deregulation' and the emergence of 'universal' banking; transformations in employment legislation and the emergence of flexible working patterns; changes in the nature of consumerism; the growth and deployment of new theories of change management; and the development and widespread implementation of new technological infrastructures.

For Financial Services, information technology, especially systems that can facilitate group work, coordination and communication of decision making, is seen as a key element in the change to more flexible and responsive forms of organization. Collaborative work, a central feature of all distributed organizations, has increasingly moved from a 'physical' to an electronic basis with the advent of widespread distributed computing. Such developments are highlighted by the introduction within organizations of email, desktop video conferencing systems, the projected development of virtual reality environments and the expansion and use of databases of electronic documents running across internationally distributed electronic networks.

Information technology is sometimes (often) the source of considerable imaginative hype. Nevertheless, it is a commonplace that many if not most diagnoses and 'solutions' to organizational change place tremendous emphasis on technology. This emphasis appears in supporting new organizational forms for the coordination and control of work, in facilitating a greater reliance on knowledge creation and conversion; in the decentralization of organizational structures; in the creation and support of more flexible patterns of intra and extra organizational relationships; and in encouraging a greater responsiveness to the consumer.

Unsurprisingly Financial Services have been quick to adopt, develop and deploy information technology in their everyday work, using a range of IT systems to deal with their increasingly wide range of operations.

1. Database systems: of various kinds – of customers, of staff, of financial products - enable Banks to keep track of customer and staff activity so as to inform decisions about the running of customer accounts and the targeting of products and advertising; as well as decisions about the daily staffing of the bank and the remuneration of staff. In the 1990s the banks embarked on a process of centralization of activities, closing many ordinary high street branches where staff had an extensive knowledge of their customers and relocating services in large regional centres. The disappearance of 'local knowledge' meant that decisions were increasingly based on knowledge of the customer derived from the logging of activity through their account. At the same time, as banks became increasingly involved in the selling of financial services, so the view of the 'good customer' began to change – from someone who merely had a high salary to one where the 'good customer' was identified by the number of financial products they had purchased.

2. Workflow systems – Logging, recording and storing activity also enabled Banks to make informed decisions about the staffing of their different units since details of daily, weekly and monthly variations facilitated such decisions. Such monitoring, as well as devices such as the Balanced Business Scorecard, based on recording the completion of particular identifiable tasks, also impacted on staff pay, promotion and bonuses.

3. 'Expert Systems': 'expert systems' of different kinds are increasingly used within the banking systems. In part this is a simple reflection of the 'audit culture' that exists in banks and in part it is a product of an attempt to automate, computerize and control a range of activities previously carried out autonomously by bank staff such as Lending Officers or Business Managers. Another feature of the 1990s and early 2000s was the increasing

restriction on the autonomy of bank staff such as Bank Managers who saw their lending limits – the amount they were allowed to lend before needing to get official sanction – increasingly reduced. Surprisingly then, perhaps, a rash of very bad lending decisions triggered the financial crisis of the late 2000s.

4. Security, monitoring and audit systems: are also, and unsurprisingly, strong features of financial services. What is interesting about their operation is the extent to which they remain reliant on a range of human factors. So, for example, banks often employ a 'mystery shopper' to ensure that staff are following appropriate procedures. Similarly, monitoring of individual transactions and accounts is often dependent on bank staff being alerted by suspicious activity in an account, or the behaviour of the account holder – for example in money laundering.

Retrospective

Our work in this area is still current. Banks have continued their automation with the principal change being the increasing use of self-service in the form of digital banking. Associated with these digital banking systems are AI-based security systems that analyse patterns of transactions for anomalies.