COMIC

ESPRIT BASIC RESEARCH PROJECT 6225

A Computational Model of Organizational Context in CSCW

Abstract

This is Deliverable 1.3 of the COMIC project. It gives an account of a computational model for organizational context. The deliverable contains an introduction followed by the presentation of the reasearch that has been done during the last year in the project. This research is described and exemplified over nine further chapters, that focus on theoretical perspectives and fieldwork investigations as well as computational models and their application.

Document ID	D1.3
Status	Final
Type	Deliverable
Version	1.0
Date	August, 1995
Editors	L. Fuchs, W. Prinz (GMD)
Task	1.3
ISBN	1-86220-006-8

Project coordinator:

Tom Rodden
Computing Department
University of Lancaster
Lancaster LA1 4YR
United Kingdom
Phone: (+44) 1524 593 823
Fax: (+44) 1524 593 608
Email: tom@comp.lancs.ac.uk

The COMIC project comprises the following institutions:

Gesellschaft für Mathematik und Datenverarbeitung (GMD), Bonn, Germany
Risø National Laboratory, Roskilde, Denmark
Royal Institute of Technology (KTH), Stockholm, Sweden
Swedish Institute for Computer Science (SICS), Stockholm, Sweden
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
University of Amsterdam, Amsterdam, The Netherlands
University of Lancaster, Lancaster, United Kingdom (Coordinating Partner)
University of Manchester, Manchester, United Kingdom
University of Milano, Milano, Italy
University of Nottingham, Nottingham, United Kingdom
University of Oulu, Oulu, Finland

Editors of this report:

Ludwin Fuchs and Wolfgang Prinz
GMD
Institute for Applied Information
Technology
Schloß Birlinghoven
53754 Sankt Augustin, Germany
Phone: (+49) 2241-142717
Fax: (+49) 2241-142084
Email: {ludwin.fuchs,prinz}@gmd.de

Lancaster University, 1995
This report is available via anonymous FTP from ftp.comp.lancs.ac.uk.
Further information on COMIC is available from: http://www.comp.lancs.ac.uk/computing/research/cseg/comic/
ISBN: 1-86220-006-8

Chapter 1: Introduction

Ludwin Fuchs and Wolfgang Prinz
GMD, Germany

The Strand 1 activities in the final year focused on two major issues. First, the considerations of various approaches to capture organizational context as well as field studies, that investigate the introduction of CSCW technology in organizational settings, to derive a range of important implications for the development of systems. Second, the development of a computational model for the support of organizational background in cooperative systems. Furthermore the applicability of this model by different CSCW applications, such as task management and workflow support has been examined. Finally we investigated how visualisation techniques developed in Strand 4 might be applied to conceive organizational information.

The results of year three are a product of these activities, but also of various revisions and discussions, that have been strongly influenced by the observations in the field studies, especially work that has been performed in Strand 2. On the contrary the model has influenced a variety of applications, most notably the development of the Milano system, which is described in this deliverable but also the work that has been performed in Strand 4. In particular the event model for the support of awareness in cooperative systems, developed in the framework of the COMIC Shared Object Service uses the modelling paradigm of this computational model as the basis for the distribution of events.

The following results have been achieved in the third year:

- A field study that is concerned with a series of issues surrounding the status of 'models' of cooperative work and its organization.
- An analysis of organizational networks that emerge from the growing international network availability and use.
- A computational object oriented model that maps the conceptual framework onto an object-oriented information model.
- The demonstration of the applicability of this model by applications, such as task management and workflow support.

- The development of new models and systems (W-TOSCA, Milano System) based on the COMIC organizational context research results.
- The development and investigation of different models and mechanisms for visualising organisations.

These results capture and exceed the initial workplan goals. This deliverable documents these results. It is structured according to the following overview:

Chapter 2: Modelling the Organization of Work:Workflow Technologies on the Print Industry Shopfloor

This chapter examines the status of 'models' of work, interaction and organizational context in CSCW. After briefly reviewing some of the various positions which have been taken in CSCW and allied fields with respect to modelling, simulation and similar enterprises, the chapter argues that the status of models (etc) should be made a practical concern and a topic for empirical investigation rather than protracted epistemological, theoretical or methodological debate. Under this aegis, a field study is reported of an organization in the print industry, examining a workflow system introduced to the shopfloor. The indigenous methods by which members order their work are datailled and contrasted with the order provided by the system and it is described how members have attempted to accommodate the two. Although it disrupted shopfloor work, the system's use was a contractural requirement on the organization to make its services accountable. This suggests workflow systems can sometimes be seen as *technologies for organizational ordering and accountability*. We conclude that the status of models of work and organizational context needs to be understood in relation to the *purposes* of the modelling enterprise and how it might go about satisfying those. The implications of this purpose-relevant understanding of 'models' are explored in relation to CSCW requirements which, specifically, should acknowledge the exigencies and organizational status of CSCW technologies which embody models of work (such as workflow systems).

Chapter 3: Hunting the Snark? The Case for & against Organizational Memory

The third chapter initially provides a conceptual critique of the concept of "organizational memory". It briefly reviews some of the rich and varied contributions from both administrative studies and information systems concerning this topic, while at the same time noting the vagueness of the term as it is commonly used. What is of interest is the pervasiveness and perseverance of this concept across a wide range of disciplinary endeavours, which are here brought together and discussed under the same umbrella. In an effort to resolve some of the ambiguities and uncertainties surrounding the concept, and improve the current state of affairs, the, chapter provides an important re-formulation of the notion of memory implicit if not explicit in most current views, i.e. the notion of memory as a passive store, arguing instead for an active, constructive view of "remembering" that has a long, if forgotten history within psychology. In the final section, some implications of such an approach are discussed, paying particular attention to the need for empirical studies of "memories in use" and common information spaces, and expanding the domain of discourse to include sociological as well as psychological perspectives on concepts such as memory, learning, remembering, talking, etc. in the context of organizations. Such a radical re-formulation of the issues surrounding organizational memory also has serious implications for the kinds of computer "support" for this phenomenon which might be possible or feasible, which can only be touched on in this chapter, but will be extended in future work.

Chapter 4: The I*EARN: an Organizational Learning Network

A first result of COMIC was the production of "A Conceptual Framework for Describing Organizations" around the question: "What is an organization?". One of the COMIC conclusions is, that organizations are increasingly transformed from "bureaucratically controlled workplaces" to more "network learning organizations" In chapter 4 we want to go a step further and analyse what we call "an organizational network", a computer and telecommunications network with organizational roles. I*EARN is the case study for this.

Chapter 5: Work Processes, Organizational Structures and Cooperation Supports: Managing Complexity

In chapter 5 the service paradigm is used to analyze and characterize work processes and their complexity. A work process, from this point of view, is characterized by the communicative relations binding its participants and embedding their performances. The complexity of work processes is then related to organizational design issues, to the empowerment of professional skills and to the computer support systems capable of helping people to manage the complexity of their work effectively. The approach to change management that underlies this chapter renovates the Socio-technical System Design Methods, allowing them to fully exploit the potential of existing Information and Communication Technology.

Chapter 6: An Object-Oriented Framework for Organizational Modelling

Chapter 6 presents a modelling framework which provides the basis for organizational information modelling in a cooperative environment. Additionally, it provides the foundation for the information model of the organizational information system called TOSCA (The Organization information System for Cooperation support Applications), developed in the course of the COMIC project. The chapter describes the definition elements for the three building blocks of the OIS object model: organization objects, relationship objects, event objects.

Chapter 7: An Organization Object Model

Chapter 7 contains a general consideration about the modelling domains of organizational information followed by a presentation of the object class schema for each building block. Only a general outline and examples for the schema will be provided as a complete description of all object classes needed for a comprehensive organization modelling would exceed the framework of this work. The aim of the proposed schema is the provision of a toolkit that can be adopted to different organizational settings.

Chapter 8: Organizational Information and Coordination Support Systems

An organization information server is designed as a service provider within an environment of applications which support cooperative work. Within such an environment we can identify two major types of organization information server applications. The first type of applications provide direct user access to organizational information, such as query and retrieval interfaces or graphical browsers. The second type of applications use the organizational information server as a central service that provides organizational information needed for the realization of their own functionality.

Chapter 8 describes the interaction between a task management system and the organizational information server. The task management system is furthermore applied for the coordinated distributed administration of the organizational information server. The appropriate concept for the synergetic interaction between both applications is described as the final section of this chapter.

Chapter 9: The Milano System

Chapter 9 describes the main aspects of the Milano system. The three components of the system - namely a Workflow Management System, a Communication Handler and an Organizational Handbook - are sketched. A description of the system architecture points out how the enabled mail technology can provide multiple levels of openness. Directions of future research are sketched at the end of the chapter.

Chapter 10: Seeing Information in the Organization

The cooperative use of information plays a central role in many work settings. As this information becomes increasingly electronic applications require facilities that promote cooperative access to shared information. This is particularly true as users browse repositories of information. A need exists to discover new ways of presenting this information to users. It is equally important that these new facilities are informed from an understanding of the use of information. Chapter 10 turns to the use of documents as a means of understanding the use of shared information within organizations. The pertinent features of documents are outlined before a field study examining the ways in which users access shared information is presented. The results of the field study are used to directly inform the development of a novel set of information visualization techniques. Rather than focus on discovering the location of information the visualization facilities focus on finding others through the document and the sharing of the document within the context of the organizational setting.

Chapter 2:
Modelling the Organization of Work: Workflow Technologies on the Print Industry Shopfloor

John Bowers
University of Manchester, UK

Graham Button
Rank Xerox Research Centre, Cambridge, UK

Wes Sharrock
University of Manchester, UK

Introduction: CSCW Technology, Organizational Models and Work Practice

This chapter is concerned with a series of issues surrounding the status of 'models' of cooperative work and its organization. Providing some kind of *model of work* has been a pre-eminent concern in much research in CSCW. For example, Winograd and Flores' (1986) The Coordinator system provides an explicit model, presented in the form of a series of transition graphs with arcs labelled with performative verbs, of the interaction that surrounds and in part comprises the conduct of work in their view. Medina-Mora et al. (1992) extend this approach by presenting the actions and interactions of the workplace as locatable within a cycle of exchange between 'customers' and 'performers'. Bowers and Churcher (1988), while critical of many of the features of The Coordinator (in particular of its claims as a model of conversation), nevertheless offer a formalism for defining what they call 'communication structures' which are intended to capture (in some sense) the regular character of cooperative interaction within work practice. Many of these themes have been continued and adapted in the COMIC project in the development of workflow and conversation management systems (see Deliverable D1.2 and elsewhere in this deliverable).

These and related attempts are many and varied. For example, they differ in terms of the extent to which they wish to draw upon empirical studies of work and interaction in existing practices or prefer to stipulate the nature of the organization of work on conceptual or theoretical grounds. They also differ in terms of their affiliation or otherwise with broader agendas for organizational change. While each of the examples cited has a commitment towards technological development, there exist other attempts to provide 'models' of work, interaction and their organization which are intended more as analytic devices than representations to be embedded in working CSCW systems.

Naturally, contributions to CSCW research of this flavour have provoked a number of criticisms. Suchman (1994), Robinson and Bannon (1991), Schmidt (1991) and Bowers (1992) all present critical analyses of 'models', 'representations', 'formalisms' and so forth (the terms vary but the targets of criticism are comparable) in CSCW. The conclusions of these criticisms and their critical strategies are again varied but they all raise concerns about the inappropriate implementation and use of models and the rest for the support of cooperative work.

This critical background comprises much of the immediate intellectual heritage of the COMIC Project and many contributions to the Project have been concerned to clarify the status of models and formalisms (and so forth) with respect to the support of cooperative work. Indeed, a repeated emphasis in Strand 1 of the project has been that the status of models of work and interaction should be understood in close relation to the *organizational context* of not merely the work but also the modelling activity itself.

This emphasis is especially urgent as over the last few years there has been much growth in research on 'computational organization theory' (Carley and Prietula, 1994) and 'simulating societies' (Gilbert and Doran, 1994) to quote the titles of two central texts. This research is concerned to model, simulate, represent (and so forth) social and organizational phenomena using various computational means, including AI and Distributed AI technologies (e.g. Doran et al., 1994), genetic algorithms (Reynolds, 1994), game theory (e.g. Shi et al., 1994) and graph theory (e.g. Krackhardt, 1994). While these contributions extend the remit of computational modelling and simulation beyond the concern for individual action and cognition (which still comprises the mainstream of AI and Cognitive Science) to an interest in social and organizational phenomena, the epistemological, theoretical and methodological standpoints espoused uniformly in the two collections cited is highly familiar. Computational modelling is championed because it can provide precise, objective and repeatable findings and hence scientific access to otherwise hidden features of social and organizational structure and process. Of course, these claims and the underlying notion of 'science' in use here are highly contestable and on very similar lines to familiar criticisms of more mainstream AI and Cognitive Science (see *inter alia* Suchman, 1987; Coulter, 1983; Dreyfus, 1979). It is for these reasons that work in Strand 1 of COMIC has had to be careful in articulating a status for 'models of organizational context' which need not trade on such commitments.

Whatever the epistemological, theoretical and methodological status of computational simulations and modelling, of models of organizational context, and of models and representations of cooperative work itself, it remains an open question whether such models have practical utility. It is quite possible for conversation management and workflow systems to have practical utility even if some of the more ambitious theoretical stories told about them are highly suspect. Equally, it is possible for some model of an organization to be a valuable aid in reasoning about important organizational decisions even if it cannot be legitimately regarded as giving 'scientific' access to the nature of the organization itself. In short, and as repeatedly emphasized in the COMIC Project, models (and the rest) should be assessed in terms of their status as *resources for practical action* and not against the yardstick of some bogus conception of science and what the social and computational sciences should be about.

This practical (re-)orientation on research issues in CSCW is central to the constructional, engineering approach advocated and developed in Strand 2, as well as to the 'mechanism of interaction' concept developed in Strand 3. However, we feel that it is important to clarify these matters and exemplify the difference that understanding models as resources for practical action makes to understanding the relationships between technology, work and organization. Accordingly, we wish to take a practical, empirical turn on questions surrounding models of work and organizational context. In this way, we hope to clarify the status of the computational work which is presented in the bulk of this deliverable. As suggested above, we feel this is urgent lest the computational work of this Strand be misunderstood as partaking in the same epistemological commitments as the recently fashionable work on computational organization theory and simulating social phenomena rather than as a practically oriented contribution to the support of cooperative work.

Workflow Technologies

To address these matters, our chapter reports a field study of workflow technology in a nationally distributed organization in the print industry. We concentrate on workflow technologies because building appropriately nuanced workflow systems has been central to the technical development work of this Strand. By 'workflow technology' we understand any technology designed to (in some way) give order to or record the unfolding of work activity over time by, for example, providing tools and information to users at appropriate moments or enabling them to overview the work process they are part of or to design work processes for themselves or others or whatever. As we have suggested, the development of workflow technology has been an important part of the research endeavour within CSCW (Medina-Mora et al., 1992; Agostini et al., 1993) arousing much controversy and debate (e.g.

Suchman, 1994; Winograd, 1994). However, not all of the discussion of workflow technology has been directly informed by empirical studies of cooperative work or of the use of emerging systems. This exposes such systems to the objection that they are insensitive to the contextual details of work and interaction in ways which inhibit their usability. Indeed, an increasing number of case studies of workflow and groupware technology in CSCW point to this as a critical problem when such systems are implemented in actual organizational contexts (e.g. Bowers, 1994; Orlikowski, 1992).

Our study extends the literature of empirical studies of CSCW technologies in use by examining a workflow system in the print industry. To date studies of technology in use in CSCW have been dominated by office work in administrative and managerial sectors. Indeed, with a few exceptions (e.g. Hughes et al., 1992), 'the office' constitutes the default development context for much of CSCW research. To us, this is an unfortunate narrowing of the horizons of CSCW and one which has influenced the debate over workflow systems. The presence of computers in office work, from familiar word processing and database systems to local area networks and so forth, is ubiquitous. The personal workstation is a prominent tool nowadays in doing office work. This makes it easy for designers to consider adding to the functionality of office systems with the introduction of workflow applications. Indeed, in some of the most developed visions, workflow will become less another application more an 'embedded enabler' of work invisible to the user much like many of the features of today's workstation operating systems (cf. Abbott and Sarin, 1994; and compare with the goal that computational mechanisms of interaction as developed in Strand 3 should be embedded within operating systems and not regarded as separate applications). However, it is important to note that this vision somewhat depends upon the ubiquity of computing as the tool and medium for the work. While this may be becoming the case for much administrative and managerial work, not all work has this character. Indeed, the existence of special purpose tools and materials, not necessarily computational or informational in nature, is precisely part of what makes production and manufacturing work distinct.

For example, on the print industry shopfloor, while computer controlled digital reprographics technologies are becoming more prominent, older technologies such as lithography and 'hot metal' presses are still deemed essential for various forms of work. In any case, most contemporary print shops are likely to contain a heterogeneous suite of technology so as to maintain flexibility in how jobs can be done. Furthermore, reliance on one form of production technology can too closely shackle a company to a small number of suppliers. Finally, customers themselves may specifically demand that a job be done one way rather than another. Thus, currently, in much of the print industry, the vision of ubiquitous workflow technologies is someway off.

This means that workflow technologies in such industries need to be at least in part *external* to the tools and materials of the work. While an office based workflow system can take the approach of providing information and computer based tools as and when they are appropriate, it is not feasible to expect a workflow system to set up a litho-machine or move a comb-binder onto the shopfloor as and when they are required for the job! In short, we feel that much of the debate surrounding workflow has been specific to *internal* workflow technologies where the tools and materials are not different in kind from those used in the work itself.

Of course, those who criticize workflow systems are often precisely objecting to the internal positioning of the system within the work. It is this, some would argue, that allows workflow systems to overly constrain the work by imposing some process model or theory of interaction on it. If a workflow system were external to the work and did not directly control the availability of resources for the work, then perhaps more flexible support for cooperative work could be offered through workflow. In some respects, this is the approach taken by De Michelis and Grasso (1992; and see their contribution to Deliverable D1.2) who present a system which separates out support for the negotiation of the nature of the work from support for communication about it.

By going beyond office work into an industrial production context for our fieldwork, we hope to subject such claims to examination while adding to the corpus of empirical studies of workflow systems and bringing to the attention of the CSCW community the nature of cooperative work on the shopfloor of the print industry.

Fieldwork in the Print Industry

Our fieldwork has been conducted at an organization which we shall call 'Establishment Printers' (EP). In this section of the paper, we shall first give an account of EP, some recent significant organizational changes it has had to face and the general nature of our fieldwork at EP's sites. Following this, we shall describe the particular sites within EP (and the particular kinds of print work they undertake) which form the focus of our studies of workflow technology.

'Establishment Printers'

EP is a nationally distributed printing company in the UK, employing a total of over 2,000 personnel. It engages in a variety of forms of print production work from large run printing of books and pamphlets to smaller scale reprographic work. As well as offering its own print production facilities, EP maintains a print brokering service whereby it liases between customers and third party providers of print services so as to negotiate the best deal for its customers. EP also offers technical advice to customers on a wide variety of issues concerned with publishing. Many of its sites offer customers origination capabilities, where EP personnel will collaborate with customers on the production of original print copy, including basic graphic design and art work. EP has in the region of 20 production sites distributed throughout the UK. In addition, EP has a small number of retail outlets where some of its own publications and those of related publishers can be purchased by the general public.

Since March 1994, we have been conducting fieldwork at EP. Our fieldwork has been concerned with the technical and organizational problems faced by large-scale print production as an instance of cooperative work. The specific occasion for our fieldwork has been a series of recent organizational and technical changes at EP which have involved some significant changes in how it prosecutes its business. For example, EP has engaged in a recent re-examination of the nature of its work in an attempt to redefine what it does in terms of its 'core business processes'. While not a full-blown exercise in Business Process Reengineering of the sort discussed (e.g.) by Hammer and Champy (1993), nevertheless EP have produced explicit definitions of business processes for the purposes of, for example, British Standards accreditation through BS5750.

Furthermore, EP has sought out new business through responding to invitations to tender for UK Government printing. Many UK national Government organizations maintain their own print production capability. Several of these have undergone 'market testing' whereby a Government organization will open its own print work out for tender. EP has successfully won a series of contracts with Government organizations as a result of this process to provide and manage those organizations' reprographic facilities 'in-house' on the customers' own sites. 'Facilities Management' or FM work, as it is known, now comprises one of the most important, profitable and expanding commercial activities EP engage in. At the time of writing EP have won 4 FM contracts, taking in 7 new print production sites to the organization.

Finally, EP has introduced a suite of technical changes into their printing work. The most important of these are based in digital reprographics technology. Indeed, at most of the sites we visited, site managers informed us that increasingly they were using high-end photocopiers and digital reprographics machines which offer scanning, storage, reproduction and networking functionality in preference to traditional off-set lithography and allied printing techniques.

While we do not have space to account for this change to digital technology at EP here, it - together with the organizational and business changes we have just mentioned - led us to specifically study the work at reprographic sites (especially Facilities Management sites) as, we conjectured, this is where the inter-relations between such changes might be most readily available for study. Accordingly, since gaining access to EP, we have visited 13 of its sites, including 5 FM sites, 7 other reprographic sites and EP's administrative headquarters. At each of these sites, we initially conducted interviews with key personnel on the shopfloor as well as in a site's administrative office to gain an overall impression of EP's business and the organizational and technical issues it involves. Following this, we have spent extensive periods on site observing work being done, attending meetings and so forth at selected sites.

Facilities Management (FM)

This paper concentrates on the use of workflow technology at the FM sites which are maintained by EP at a UK Government organization which we shall call 'The Department'. EP, as a result of a market testing and competitive tendering process, undertakes The Department's print work at the Department's own sites. In line with the regulations governing the market testing process, EP has taken over the existing reprographic equipment and personnel at The Department's two facilities in the North of England and one in London. The contract between The Department and EP is based on the level of service specified in the invitation to tender and the means by which EP offered to meet it in their own tender. It is for a fixed three year term and runs until 1998 when the market testing and tendering process begins again.

Importantly, the contract binding EP to The Department not only details the level of service to be provided but also specifies how EP's actually delivered service will be monitored. Indeed, EP faces penalties under the contract if their delivered service is found not to match the level required and promised. In particular, as one of the means for monitoring the service provided, the initial invitation to tender required that applicants describe the Management Information System (MIS) with 'shopfloor data-capture' which they would install if their tender was successful. After commissioning a consultant's report, EP committed themselves to install a workflow system, which we shall call 'PRINTFLOW PF2', to meet this requirement. PF2 embodies a model of print work which contains a number of commitments about its supposed nature and organization. It is this system, its embodied model and its impacts upon cooperative work on the shopfloor of EP that we shall describe.

As another means for monitoring EP's service, The Department each month conducts a 'mini-audit' by randomly selecting 20 orders for print work out of the many hundreds which EP will have received. This sample is then sent back to each of the employees of The Department who initiated the order for print work as a check that the job was indeed completed by EP according to specification, within agreed deadlines and so forth. Naturally, The Department are concerned to ensure that they do obtain a high quality service from this new 'outside source'. However, The Department are especially attentive in case EP turn out not to be up to the job so that some other contractor might be successful the next time The Department's work is market tested. After all, The Department themselves were one of the unsuccessful tenderers for their own print work last time!

Reprographic Work at 'Establishment Printers'

Before we enter into the details of PRINTFLOW PF2 and its usage at The Department, it is important to appreciate some general features of EP's reprographic work. By 'reprographic work', we mean all forms of printing and copying which use digital copying, photocopying or off-set lithography as the principal printing tools for the job. Reprographic work takes in all small to moderate scale copying work rather than very large print run jobs (i.e. hundred thousands or millions of impressions) which may be completed, for example, on traditional 'hot metal' presses.

The reprographic work we have studied at EP is very varied. EP will undertake jobs involving copying a single sheet of white A4 paper just once up to the production of bound booklets in a 10,000 print run. A single site, which typically will have of the order of 20 staff in total, often will have the expertise and technical capability to undertake a variety of forms of copying, printing and reproduction, together with different kinds of finishing, binding, wrapping and packing. Some of EP's sites also offer their own collection and delivery service. A site may also have its own Desk Top Publishing capability and offer this as a service to its customers. Jobs may come in from customers as hard copy or on disk in a variety of formats or as an ASCII file over a modem or as even, say, a hand drawn specification of a poster with some sketches which the DTP personnel will work on in negotiation with the customer. Some jobs can be initiated direct from the hard copy that customers provide. In other cases, the hard copy might require completion or correction and further proofing by EP's own personnel or (even) 'cleaning up' through digital scanning and masking over blemishes.

A job, then, can take many different forms, involving different volumes, production techniques, materials, skills and so forth. Jobs can also run on quite different time-scales and require scheduling in quite different ways. At one extreme, some of EP's work is known in advance as is the case, for example, with a report that EP print every month by agreement with a customer. When it is due and even some of its content (e.g. a standard binding or cover sheet) can be anticipated. EP also offer a ten day turnaround service for any scale of reprographic work. Furthermore, at two of The Department's FM sites, for example, EP offer a same day counter service for smaller scale jobs. In addition, EP engage in some 'print-on-demand' work where they store master copies (typically in digital form) on behalf of clients and print out small numbers of impressions to order as and when the customer requires them. Finally, to maintain good relations with important or highly valued customers, EP are quite willing to accept highly urgent work and give an immediate turnaround on it.

In short, then, a reprographic site at EP faces great variability in the kind of print work they may be required to do. They may receive little or no notice of work which may itself have varying degrees of importance to the customer and urgency. Furthermore, very little of the scheduling of when jobs come in is under the direct influence of EP themselves. Indeed, as EP has undertaken to diversify its services and to adopt a 'customer centred' business philosophy in recent years, the problem of managing the work has become especially acute. How are these problems to be dealt with? How is the work of the reprographic sites to be ordered so that EP can provide the level of service its customers expect? How can the massive contingencies of print work be managed?

To a certain extent, known-in-advance work can be used to provide a 'grid' around which other jobs can be fitted. Thus, a particular machine and/or operator might be assigned to an expected job in anticipation of it. However, at most sites (and especially the FM sites where a counter-service is offered), it is the contingent jobs which makes up the bulk of EP's work. Accordingly, ordering the work around know-in-advance work can be at best just one management device amongst others. Furthermore, as FM sites constitute a ready-to-hand and visible presence of a reprographic capability, it is an all-too-familiar occurrence for provisional plans to be disrupted by a large scale and highly urgent job delivered in person.

Maintaining a Smooth Flow of Work

If advance planning and allocation of jobs to machinery and personnel is made highly problematic by the contingency of the major bulk of EP's jobs, one has to look elsewhere for the key to how the complexity of reprographic work is managed. Our claim is that, in essence, reprographic work is managed by ensuring *a smooth flow of work* is maintained through all parts of a site's capability. A smooth flow of work involves ensuring, for example, that no one operator is conspicuously occupied while others are idle, that no one job needlessly ties up the shopfloor while other jobs are waiting, and that machines are appropriately used to their best capabilities. In short, a smooth flow of work consists in ensuring the even distribution of work across operators, machines and jobs.

Crucially, EP personnel will organize their work so that there is a smooth flow of work *through the shopfloor* as jobs typically spend most of their time there rather than in the 'front office', in DTP or awaiting dispatch. Hence, time delays there are likely to be more critical. Indeed, people in the front office (who open the mail, process new job orders, deal with customer queries by 'phone and fax and so forth) are themselves attentive to the nature of shopfloor work, often having worked there themselves, and allocate jobs with the smooth running of the shopfloor as a major consideration. In a sense, then, organising the flow of work through the shopfloor is central to organising the flow of work through the site and back to the customers, in short, to managing the site.

We are not offering this notion of 'the smooth flow of work' as any kind of theoretical abstraction on our part, for it is precisely in these terms that several site managers themselves articulated 'what it is that I do': 'ensure the work flows smoothly'. In our observations, we have identified four major ways in which smooth flow is accomplished on the shopfloor:

- prioritising work
- anticipating work
- supporting each other's work and knowing the machines
- identifying and allocating interruptible work

and it is these that we shall now explicate in turn.

Prioritising Work

On receipt of an order for print work, administrative, front office staff at the site will enter the details of the job (materials required, cost code, numbers, desired delivery dates etc.) into their records. A copy of the order is then used as a 'docket' which accompanies the job in a transparent jacket as it undergoes the various printing processes appropriate for it. Administrative staff allocate the jobs to shop floor workers, depending on (i) the kind of work it is (colour work will go to the operator skilled in, say, colour lithography) and (ii) who is currently most able to do the work (this is often the subject of a degree of negotiation between shopfloor and administrative staff). Typically, jobs are then placed into an in-tray located near the machinery which the first process on the job will be executed on. Once the first process (say colour lithography) has been completed, the operator will consult the job docket and pass the job on to the operator who is most appropriate (either in terms of their skills or availability) for the next process (say comb binding), again possibly after some discussion, again depositing the work in the relevant in-tray. And so forth.

Typically, jobs will be sorted or inserted into in-trays so that date-order is maintained. That is, the work with the soonest delivery time will appear at the top. However, operators do not always process work in date-order. Operators will juggle the contents of their in-trays to ensure that a smooth flow of work will be promoted. Operators will examine job dockets to make assessments about how complicated a job is and how long it will take, whether there are further time consuming processes that the job will encounter later on and so forth - these considerations being balanced against the customer's delivery date. This re-ordering will also be influenced by whether there are outstanding jobs from the previous day's work, whether a regular job is imminent and so forth.

The use of advanced digital reprographics technology makes it especially important that operators carefully schedule their work and do not blindly follow date-order. For example, a job involving several thousands of copies of standard grade white A4 paper may only require the operator to make periodic checks on the functioning of the machine and copy quality while replenishing paper. In the mean time, the next few jobs can be digitally scanned into the machine's memory and, if necessary, cropped to size or blemishes can be masked. On the digital reprographics machines at EP, scanning and copying can be executed independently. In this way, a skilled operator can set the machine off with a large scale routine job while simultaneously engaging in the more labour intensive processes of scanning, cropping, masking and so forth. Accordingly, operators will search their in-trays for just such routine jobs and juggle the rest of its contents to enable the machine to be kept in continual and smooth production. Finding a large scale routine job also helps keep digital reprographics machines in continual use over lunch and other breaktimes. One operator will set a long job in train and take a break with another coming back from their break to complete the work. The identification of such jobs within the in-tray and ways of covering each other accordingly is essential for the effective and smooth use of the machine. Thus, very rarely will strict date-order happen to coincide with the ordering that makes for a smooth flow of work.

Anticipating Work

As we have remarked, regular, known-in-advance work can be used by staff to provide a 'grid' around which other work can be fitted. While this certainly does not provide a complete solution to the problem of managing the complexity of print work, the fact that some of EP's work is anticipatable enables shopfloor workers to ready themselves and their machines for when the job does properly arrive. Indeed, some parts of a regular job like a monthly report done to a standard format may be started even in advance of receiving the month's contents. Not only can the materials can be ordered in advance, perhaps the front and back cover and standard introductory material can be printed if they are identical month-by-month. In other words, to ensure a smooth flow of work, EP staff will maximize the benefits of scheduling work around known-in-advance work by 'jumping the gun'.

Print-on-demand work especially lends itself to jumping the gun. As the originals are stored digitally by EP themselves, no setting up is required, merely the production of a specified number of copies. Indeed, some print-on-demand work may involve a fixed number of copies to be produced weekly or monthly. Under such circumstances, EP may often produce copies in anticipation of the order being confirmed. What is more, customers in The Department may merely 'phone through the order rather than themselves complete an order form. An order form may then be completed retrospectively or the details may be inserted into EP's records from a memo taken by EP's own staff at the time the phonecall was taken. Accordingly, by jumping the gun both administrative and shopfloor staff can invest print-on-demand work with the character of regular, known-in-advance work and use this as a way of promoting the smooth flow of work through their hands.

Supporting Each Other's Work and Knowing the Machines

We have already remarked that administrative staff are attentive to the workloads of shopfloor workers and are able to take this into account when allocating work. This awareness is obtained through periodically walking through the shopfloor and checking everything is satisfactory, by discussing work explicitly when new jobs are to be allocated, by chatting at break times and so forth. Equally, shopfloor workers are attentive to each other's workloads and are able to take this into account when negotiating with administrative staff or judging whom to pass work on to for its next production process, when there is more than one relevantly skilled worker.

Importantly, various architectural, machine lay-out and other ecological features of the shopfloor support this monitoring of each other's work. At each of EP's FM sites with The Department, related machines are clustered together and distributed so that jobs tend to migrate in a known direction as they are completed. For example, at The Department's London site, jobs on leaving the administrative office tend to traverse the shopfloor in an anticlockwise direction. In this way, a glance around the shopfloor will enable both shopfloor workers and administrative staff to montior the flow of work, checking on who is at their machine or absent because they are fetching paper, taking a job to the next process, to dispatch, to the administrative office, taking a break or whatever. Furthermore, the clustering of related machines not only enables, under some circumstances, one person to operate two machines at once, it also promotes the awareness operators have of each other's workloads, whether anyone needs help and so forth. In short, the ecology of the shopfloor provides lines of sight between workers which enable their awareness of each other's work and which support ad hoc cooperation.

In addition to the resource provided by a direct line of sight, operators are able to *hear* whether the work is progressing smoothly. Many of the machines have 'designed-in' alerts (beeps and so forth) which draw attention to paper jams or empty paper trays. These sounds are available, of course, not only to the operator of the machine but to anyone else within earshot. Accordingly, operators can help each other out by replenishing paper if, say, the operator who initiated a job is taking a comfort break at the moment. Furthermore, skilled and experienced print workers are attentive to the regular noises a machine makes at different stages during production. For example, a change in pitch can inform an operator that a paper-tray is about to become exhausted, or a noise of a certain sort might suggest an obstruction in one part of the paper path. In short, both an awareness of the state of the flow of work through the shopfloor and ad hoc cooperation are sustained through listening to and knowing the machines.

In summary, print workers engage in a number of activities through which they are able to monitor the workload of others and themselves and the functioning of others' and their own machinery. Not only do these activities acquaint workers with whether the flow of work through the shopfloor is smooth or not, they provide them with resources for the immediate and ad hoc remedying of irregularities by, for example, helping someone out.

Identifying and Allocating Interruptible Work

We have remarked that very little of the workload of EP's FM sites can be thought of as known-in-advance work and that even if the benefits of known-in-advance work with respect to ensuring a smooth flow can be maximally exploited by jumping the gun, most of the flow of work has to be 'ad hoc-ed'. Knowing that interruptions by highly urgent work and on the spot rescheduling are endemic to print production at FM sites, EP's staff have evolved ways of allocating and re-allocating work to ensure that such occurrences have as minimal an effect on the work's smooth flow as possible.

We have already seen that local, ad hoc cooperative arrangements will be made by operators to help each other out in times of trouble. A job which turns out to be more labour intensive than anticipated by administrative personnel may be split between two operators by local agreement on the shopfloor. Furthermore, the workers who are normally dedicated to the counter service may be utilized if a job passing through the shopfloor requires extra hands. Counter service work as we have remarked is subject to daily and weekly fluctuation. Indeed, there can be moments when the counter staff are overworked, in which case staff from dispatch or even the administrative office will 'turn a hand'. But equally, there can be idle moments. Accordingly, knowing this, administrative staff allocate counter service personnel further print jobs but ones which tend (i) not to be urgent, (ii) not to require special materials or techniques not available on the photocopiers in counter service; and (iii) to be interruptible in the event of new counter service work coming in. In short, simple, non-urgent, interruptible jobs will also be allocated to counter service workers to be 'taken up and put down' as and when there are slack moments on the counter.

Not only can administrative staff more fully utilize counter service staff by identifying and allocating them interruptible jobs, shopfloor operators may also pass appropriate work to counter service personnel when the workload on the shopfloor is becoming intense. At one of The Department's sites in the North of England, a large window has been positioned between the counter service area, where Mary works, and the shopfloor. This enables both parties to check on the workload of the other and detect moments when the flow of work is unevenly distributed. At times, no more than a raising of eyes to Mary and an agreeing nod will ensure that work is quickly passed over. Indeed, as Mary's non-counter work is designed by the administrative staff to be interruptible, she is also able to help the shopfloor workers if required and if her or her machine's capabilities are appropriate. Thus, 'passing work to Mary' (or her equivalents) becomes a further means for ensuring the smooth flow of work.

The Promise of Workflow Technology

The practices we have identified constitute the indigenous means by which staff at EP's FM sites organize and give structure to their working day. As far as we can ascertain, these practices have a long history and were utilized in The Department's print shops before they were taken over by EP. However, EP winning the contract to provide printing facilities to The Department had the major implication that a new information system should be overlaid on the work so that it could be closely monitored by The Department. Essentially, PRINTFLOW PF2, the MIS proposed and installed by EP, has two components:

- *An administrative component* in which jobs are registered in terms of their type, customer, cost code, delivery deadline and so forth. A job name will be assigned to the job by the member of administrative staff registering the job. Often this will be a truncated version of the customer's description of the job on the order form. Specifying the type of job it is involves the entry of how the job will be executed in terms of a serial model of the processes it will go through (e.g. copying, binding, finishing etc.). This information is stored in a database which can be searched in various ways and from which periodic reports can be extracted.

- *A shopfloor component* consisting of a number of shopstations arranged around the shopfloor and locally networked to the administrative database. The shopstations consist of a series of keypads. Some of these keypads are specifically configured with the names of workers and labelled accordingly. Others refer to the machines in the vicinity of the shopstation and are also so named. Others name the processes which can be executed by the machines near the shopstation and are used to notify PF2 whether an operator is starting a process or has just completed it. Others name materials which are commonly used by the machines in the execution of their processes. Yet others are used to control an operator's interaction with the shopstation and are labelled 'enter', 'yes', 'no' etc. A numeric key pad is also available. Running across the top of the shopstation is a single line, 20 character display which echoes input and illuminates with system messages while the shopstation is being used.

It is by interacting with the shopstations that shopfloor data is 'captured' by PF2. On starting a printing process, a typical interaction with a PF2 shopstation would involve the following actions in response to prompts and other information given on the display:

(1) Pressing a button named 'Start process'.
(2) Pressing a button labelled with the operator's name.
(3) Entering a job number from the docket using the numeric keypad. Pressing 'enter' when done.
(4) The job name is then retrieved from the PF2 database, displayed and 'enter' is pressed to confirm that the job has been correctly identified. The display returns to showing the date and time as it did before step (1).

On completing a printing process, a typical interaction would involve:

(1) Pressing a button named 'End process'.
(2) PF2 would retrieve the operator's name, which would then be confirmed by pressing 'enter'.
(3) A button named 'normal' is then pressed.
(4) A button named 'material' is pressed. A code (or specially configured keypad for standard paper like white A4) will be entered representing a kind of paper (there are separate codes for different colours, sizes and weights).
(5) A button named 'pull material' will then be pressed to confirm materials selected. This will then cause PF2 to ask the appropriate questions about the material used. (Thus, steps (6) to (8) may be executed differently for different materials types.)
(6) A specially configured button (e.g. labelled with the machine's name) is pressed to indicate the machine used.
(7) The number of copies is then entered using the numeric keypad.

(8) 'Enter' is pressed to confirm.
(9) Steps (3) to (8) may be repeated if there are more materials to record.
(10) In the event of wastage, instead of pressing 'normal' at step (3), a button labelled 'waste' is pressed and numbers of each kind of material wasted in the process (e.g. through machine malfunction or operator error) would then be recorded by analogy with steps (3) to (8).
(11) When all materials used and wastage have been recorded, 'send' is pressed. The display briefly shows 'Transmitted to computer' before returning to show the date and time.

PF2 was selected because it seemed to have a number of important benefits for EP. As they have been explained to us:

- it was specifically designed by its manufacturers for use on the shopfloor of the print industry
- it was consistent with the requirements of the invitation to tender and specifically would enable EP to provide management reports which could detail the time spent on processes (PF2 computes this on the basis of when operators notify starting and finishing them at the shopstations), materials consumed, wastage figures and both support accurate invoicing as well as provide, if required in the monthly 'audit', the justification (in the form of a job report) for the charge made
- it could, in principle, provide support for stock control, through keeping an accurate and up-to-the-moment record of materials used which can be inspected at the administrative component
- it can provide a record of worker activity and hence could, in principle, replace clocking on/off
- it can support process management by giving administrative staff and site managers a view of the activity of the shopfloor which can be sampled from the administrative component at any moment
- as the PF2 databases had been networked across all three of The Department's FM sites, cross-site monitoring would be supported so that sites can be aware of what each other are doing in case, say, a big job has to be shared between more than one site
- as jobs are registered and their execution is recorded on PF2 as a series of print processes, this reinforces the quality standards which EP subscribe to as this is how they depict print work too.

The promises of PF2, then, were many and varied. However, of all these, of course, the most important is that PF2 can provide reports of various sorts on how the work has been done by EP, so that EP can demonstrate to The Department, when required, that they are indeed conducting The Department's print work according to contract.

Workflow Technology: Disrupting Smooth Flow

In various ways, the introduction of PRINTFLOW PF2 itself disrupted the smooth flow of work through the shopfloor of the FM sites, raising a series of problems about how the technology should be used and whether its use could be legitimately 'suspended' on occasion. At the time of writing, nearly a year after the commencement of EP's contract with The Department, a number of the dilemmas surrounding the use of PF2 in relation to the print worker's indigenous practice for organising their work remain unresolved. Let us give some details.

The Imposition of Procedure

We noted that print work at The Department often requires jumping the gun. Urgent jobs often need to be started before order forms have been received or job numbers issued. Similarly, much known-in-advance work and regular, print-on-demand work needs to be started in advance of orders so that, amongst other reasons, more contingent work can be flexibly structured around them. However, a job for which no job number yet exists has the same status as a non-existent job as far as PF2 is concerned: no details of its execution can be recorded at the shopstations because there is no job number to enter. Accordingly, just how jobs which have jumped the gun have actually been completed cannot be made visible using PF2 conventionally. This raises some dilemmas. For example, independent records could be manually kept for how such a job is done and its details entered some time later. However, in this case, PF2 would give inaccurate information for process management and cross-site monitoring, out-of-date stock control information, as well as indicate some timing anomalies which could be detected by The Department. Alternatively, one might not use PF2 at all for such work. But then EP would be seeking to charge The Department for work which, according to the record, had not been done, while keeping records independently from PF2 would risk defaulting on the contract. Finally, counter service work is especially problematic in this regard as PF2 would seem to require the registration of administrative details before the copying service can be completed. While one solution would be to have all counter work set up as just one 'rolling' job, this would not allow separate billing for each customer according to individual customer codes. In short, PF2 imposes a procedure on the work (first register it, then do it) which negates important means for ensuring the work's smooth flow and timely delivery.

Work as Processes in Series

Shopfloor workers soon discovered a major constraint on how PF2 could be used to record data on the execution of print processes. PF2 embodied a process model which depicted jobs as consisting of a series of processes, each the responsibility of just one operator, each to be terminated before the next one commenced. Indeed, as PF2 was proposed as a means for capturing accurate data about shopfloor workers as well as about jobs, it had implemented the constraint that a single operator cannot engage in more than one process at a time. This means that the organization of the smooth flow of work accomplished by juggling in-trays cannot be made visible to the system. For example, we noted that operators of digital reprographic technology with separate scanning components need to order their work so that a long job which can run autonomously can be scheduled alongside one or more labour intensive scanning jobs. However, once PF2 has been notified of the commencement of printing the long job, the same operator cannot notify the system of any of the scanning jobs. Again there is a dilemma. Either workers accept PF2's constraints and cease juggling their in-trays or they continue to juggle their workload and not log-in the scanning process. Both of these upshots are problematic: the first because it disables an important means for ensuring smooth workflow, the second because it makes invisible to The Department something which had actually been done and which EP would like to separately itemize in billing precisely because of its labour intensive nature.

The Overhead of Use

There was a considerable overhead incurred in using PF2 to log the progress of work on a job on the shopfloor. As should be clear from the listing given above of the actions operators have to engage in to give data to PF2, an interaction consists of much keypad pressing. Indeed, a job involving several materials with non-standard codes and some wastage may require several minutes to register. This, when an operator may get through tens of jobs a day, is a noticeable overhead. Furthermore, the time spent using PF2 is not itself calibrated with the time spent on the job. Twenty thousand copies takes approximately the same time to record as two copies. Small scale print jobs which happened to use three colours of paper would require about three times as many key presses at the shopstation than a big job using one colour of paper. And so forth. Using shop stations, then, was proportionally a very big overhead for small jobs. This problem was especially acute for counter service work, when the use of PF2 actually disrupts the service counter customers receive as they have to wait a little longer while job details are inserted (even assuming appropriate customer codes can be found for them!).

The Individualization of Work

PF2 conceives of print processes being the responsibility of single operators. However, this means that the contingent cooperative activities so important to organising a smooth flow of work on the shopfloor cannot be represented to the system. If a big job were done on a digital reprographics machine, say, by one operator replenishing the paper, while another unloaded the copies from the stacker, while a third scanned in the next ten pages, they would have to discuss amongst themselves which of the three should appear as having done all of the work and which would be the two who would potentially appear as 'idle'! Equally, if a job was taken over by another operator to maintain smooth and continual production over lunch breaks, the second operator would have to 'impersonate' the first to be able to register the job done at all, thereby making the first operator's lunch break itself invisible! Again, PF2 presents problems as to how it is to be incorporated into print work without either dismantling the ad hoc practices which promote smooth workflow or falsifying the record in a way which is potentially detectable by The Department. Either way, what is accomplished by monitoring each other's activities is invisible to the system

Re-Establishing Working Order

How did the staff at EP cope with these problems? It is first worth reminding ourselves that they do *have to* solve them for all practical purposes because working with PRINTFLOW PF2 is necessary due to contractual requirements. EP *cannot* discontinue the technology even though its serial, individualized model of the organization of print work seems inconsistent with the practices by means of which print work actually gets done. Accordingly, they have to re-establish the order of their work with it (or in spite of it). This, interestingly, has been accomplished by different sites in different ways. At The Department's London site, some built-in workarounds within PF2 were discovered and these have been put to use. In contrast, after experimenting with PF2's own workarounds, The Department's sites in the North of England took a more drastic step and reorganized their entire working day to accommodate the system. Let us take these in turn.

Some aspects of the 'overhead of use' problem can be addressed through a facility called 'gang job'. This involves using the shopstations to define a 'gang' of jobs. While each maintains its own customer code, materials used can be registered in a single total per material type. Thus, if there are a large number of small jobs all using white standard grade A4, all the copies made can be returned as just one overall total. While a little effort is required to define the gang at the outset of the process, much time is saved at the end as separate figures do not have to be entered for each job. However, this workaround leads to anomalies. In order to assign operator times, materials and wastage to each job, the total time and the rest are divided through by the number of jobs and equal numbers counted to each job. This will mean that a two copy job and two hundred copy job, if they are part of the same gang, will be equally recorded when there is a hundred-fold difference in materials between them. Furthermore, if the gang involves a range of different materials, a job which consisted of only white paper may (absurdly) be recorded as having 0.09 sheets of blue included! Clearly, there is a further dilemma here. Easing the burden on the shopfloor may lead to inaccurate management information to present to The Department and potentially inaccurate billing. Avoiding this dilemma means that 'gang job' can only be used marginally: e.g. when a number of highly homogeneous jobs do happen to turn up at the same time.

Some aspects of the problems of individualising processes can also be addressed by recording a process as a 'labour charge'. This can be done by an operator even if they are engaged on another process which they have initiated by interacting with PF2 as described above. To record a 'labour charge', an operator presses a button on the shopstation marked 'labour charge', identifies themselves, types in a job number, confirms the title when PF2 retrieves it, enters a time in hours and decimal parts of an hour, followed by the details of materials and wastage as before. At The Department's London site, this facility is used to workaround the constraint that an operator cannot start up a new process on PF2 while another is active. However, this workaround does have its own problems. Labour charges can only be recorded *after* the process has been completed, so they do not support any moment-by-moment process management or cross-site monitoring and require operators to keep an independent record of when they started a job. Furthermore, whether something is recorded as a labour charge or as a timed process depends (arbitrarily) upon what happens to interrupt what in the unfolding of an operator's work and not necessarily in terms of any other feature of the job.

An alternative solution to the problem of the individualization of work was attempted at one of The Department's sites in the North. Processes which interrupted or which were run concurrently with timed-processes were recorded to a fictitious operator! When management reports of the activities of the FM site were prepared, the time allocated to the phantom operator was then re-assigned to one or more members of the actual, incarnate workforce. However, this solution raised a further problem because there was the risk of visibly assigning hours to operators who were already recorded as working full-time. Indeed, the concealment of the phantom operator's time within the hours actually worked without making any of the real workers seem super-human turned out to be difficult, not to say contrary to the spirit of arrangements between The Department and EP. Accordingly, this workaround was soon abandoned.

At the time of our last visits to EP's FM sites, the London site was persisting with using PF2's 'labour charge' facility as best they could to record concurrent work, keeping independent, manual records when necessary (e.g. if the gun is jumped) and encouraging the workforce to be as accurate as possible in recording materials and wastage. In this way, a momentary peace has come into existence between the indigenous practices for maintaining smooth workflow and the demands of using PF2. This 'peace' though has come at the cost of working overtime every week since the FM contract began and of only 'clearing the in-trays' during the working day once in six months. And that was after PF2 had been 'down' for three days!

In contrast, the northern sites have developed a more radical solution. Here, jobs are entered at the beginning of the working day into the administrative, database component of PF2, as intended. However, the system is not used to enter details of how the job was done in the course of working on it. Rather, paper based, manual records are kept on the paper dockets which accompany the job and, where necessary, further scrap paper notes are inserted into the transparent plastic jacket. The shopstations remain untouched and the flow of work is organized by means of the methods we have identified. However, at the end of the day or the beginning of the next, one member of the administrative staff takes all of the dockets, scraps and orders (including those associated with counter service work) and records the details of operators, materials, wastage and labour times into PF2. In other words, PF2 is no longer used to provide real-time shopfloor data capture, rather it is used a means for *retrospectively reconstructing* the day's work in a form which can nevertheless provide The Department with an account of what occurred and why they have been charged in the way they have, even though this forfeits many of PF2's other promised benefits.

Workflow and Technologies of Accountability and Organizational Ordering

Our fieldwork suggests to us that extreme difficulties can be encountered when introducing workflow systems into a workplace, even when the system is external to the tools and materials of the work itself. PF2 does not (directly) constrain workers' access to the resources for their work, nor (directly) impose an ordering on the work by insisting that some task be completed before another one. However, it does embody a process model of how print work is done which makes recording the work problematic in the light of what is actually done. This case suggests, then, that the impacts a workflow system may have on work can be extensive even if there are no 'hard wired' links between the system and the conduct of the work.

Workflow From Within and From Without

We feel that the image of 'workflow' captures well the question of the temporal ordering of work. Indeed, this is how several of EP's staff themselves characterized things when all was well: 'the work flows smoothly' and such like. Work unfolds over time and has to be organized (scheduled, conducted, recorded, managed) attending to this. The question is not whether smooth workflow is to exist or not (it has to) but how this is to be accomplished. It is in this connection that we would like to offer a distinction between workflow *from within* and workflow *from without*.

Workflow from within accomplishes the smooth flow of work through methods which are internal to the work. To do print work competently requires that, on receipt of a job, an operator is able to orient to matters such as: Is this job properly for me? Should it be done next? How urgent is it? To whom should I pass it when I am done? And so forth. By resolving these questions in working on the job or passing it to someone else, not only is the job done, so is the organization of the shopfloor in part accomplished. Workflow from within characterizes the methods used on the shopfloor which emphasize the local and internal accomplishment of the ordering of work. Workers juggle their in-trays, jump the gun, glance across the shopfloor, listen to the sounds coming from machines, re-distribute the work in the *here and now* so that what to do next can be resolved. In the here and now, in *real* *time*, operators and administrative staff encounter multiple jobs of a typically heterogeneous nature, requiring artful scheduling and completion.

In contrast, workflow from without seeks to order the work through methods other than those which the work itself provides. In PF2, a formal model of the work is provided which depicts print work as processes in series such that (i) each process has to be fully terminated before another can begin, (ii) each process can have just one operator associated with it at any one time, (iii) each operator can only engage in one process at any one time, and so forth. It would be inaccurate of us to say that these methods from without are just plain wrong. Rather, they offer *another way of organising print work*, one which is encountered by the workers at EP's FM sites as alien to (because from without) *their* methods of organising print work. Their methods crucially attend to the problem of the ad hoc, real-time ordering of multiply instantiated jobs. PF2's are concerned with the processual character of individual jobs, engaged with by individual workers, measurable by clock time and so forth.

In the case of EP's FM sites, workflow from within comes to be in tension with workflow from without as soon as the latter has to also be *reckoned on within the work*. The difficulties with PF2 arise not because some technology merely offers an 'incorrect' workflow model, nor (even) because that model is inserted into the work, but because it is inserted in such a way that makes the accountability of workers and the work that they do problematic in new ways which are themselves hard to deal with. It is because PF2 *has to be* worked with and accommodated that the tensions and dilemmas we have noted arise. These tensions can be negotiated in various ways - distributing them across the shopfloor and throughout the working day (as at the London site, where all shopfloor workers use the system as best they can at the cost of overtime and delays) or allocating them to one worker at a given part of the day (as at the northern sites where the day's work is reconstructed through someone inputting job data retrospectively). Either way these accommodations arise through the practical necessity of having to use the system. And it is this practical necessity which binds the use of PF2 to print work at EP just as strongly as any internal 'hard wiring' of a workflow system to the actual conduct of work might do.

Technologies of Accountability and Organizational Ordering

While the potential tensions between workflow systems and work has been emphasized by a number of writers, not all present empirical cases which actually display this tension (if it is experienced) and how it's worked with and accommodated (if it is). Rather arguments within CSCW have had a tendency to take 'in principle' stands on the question of the relation between certain forms of ('bad') technology and cooperative work. However, our case points us to alternative considerations.

We do not think that the PF2 case should be simplistically seen as a straightforward 'Horror Story of Bad Design' or an example of the 'violence' which can be done to workers by formalism. While we are not saying PF2 cannot be improved, it is important to be clear about the requirements that technologies like it may be trying to meet and that organizations seek to realize in implementing them. For EP, these requirements are not (at least not in any direct sense) to support cooperative work or to externalize process representations or to control the workforce by observation and new forms of discipline. It's altogether more mundane: they had to install such a system if they were to win the contract for new work at all! And as a contributory means for winning the competition for the contract, PF2, being specifically designed for the print industry, was the most prominent choice. Other requirements are descended from this.

As such, EP's position may not be altogether untypical. Often organizations have to make promises about the technologies they will install to gain new work or use the technologies they have not merely to do the work, but to advertise that they are the kind of organization prospective new customers would like to do business with and so forth. In summary, in our view, no crude morals about workflow and allied CSCW systems should be drawn without understanding as fully as one can the organizational significance of such technologies. Similarly, no simplified theoretical view of the relation between organizations, work practice and technology should be offered when *organizations themselves* reconfigure these relations in the conduct of their business or in seeking new work.

Suchman (1992) offers a concept to help articulate some of these issues: 'technologies of accountability'.

> By technologies of accountability, I mean systems aimed at the inscription and documentation of actions to which parties are accountable not only in the ethnomethodological sense of that term (Garfinkel and Sacks, 1970), but in the sense represented by the bookkeeper's ledger, the record of accounts paid and those still outstanding.

This is a useful way of understanding PF2 with respect to the relations EP need to sustain with The Department. PF2 is the required means for producing *accounts* both as records of what is charged and as documents which *visibly testify* to EP's efficiency, appropriateness, trustfulness, loyalty to contractual terms and so forth. Management reports from or based on PF2 can be used to provide not just a record but a *justification* for what was charged and what was done if EP are *called to account* by The Department.

We feel our fieldwork adds some important details to Suchman's concept. Not only do workflow systems often provide the means for the production of accounts (in this dual sense), technologies of accountability need to be understood *organizationally* and *inter-organizationally*. Introducing technologies of accountability can be in tension with existing ways of organising the work, as well as provoke or fall in line with new ways of constituting the very organization within which the work is done. For The Department, EP are not merely the organization that happen to now do their print work. They are the organization who do print work in a visible, inspectable, documentable, measurable, accountable way. This is a matter of organizational change not only for The Department (who have 'lost' their print facility under market testing) but for EP (who have not only gained business but business which has to be conducted in a new way). In all these senses, technologies of accountability can be *technologies for organizational ordering*, as part of how organizations come to be redefined through new trading relationships.

Conclusions: Models of Work and Organization

Let us return to the issues which motivated this chapter. The potential for 'mismatch' between models of work and its organization on the one hand and work itself on the other has often been noted. Indeed, as we have acknowledged, the current study, in displaying tensions between workflow from within and without, echoes this theme too. However, in the present case, this did *not* constitute a decisive reason for rejecting or objecting to the system. The match or otherwise between model and work was just one of an array of considerations which organization members needed to reckon on. Indeed, as a matter of practical contractural necessity, objections to PRINTFLOW PF2 of this order needed to be subordinated to the demand to generate reports with the system to visibly meet the level of service that Establishment Printers are committed to provide. As such, what is relevant for the appraisal of PF2 is not just whether its workflow model 'matches' work itself (whatever that might mean) but whether PF2 works adequately as (part of) a technology for accountability and inter-organizational ordering in the senses laid out above. Accordingly, in this case, the viability of PF2's workflow model hinges upon its contribution to the production of accounts (documents, invoices, management reports etc.). Thus, if one is (in this case) to negatively evaluate PF2's individualized, serial model of printing processes, it is not just because print work is not like that but also because this model has questionable utility in the production of accounts of the work which are usable by EP in demonstrating their level of service to The Department. Indeed, the real-time operation of the model is precisely what is abandoned at the northern sites where the shopstations mostly remain idle. In short, the current case suggests that models of work and its organization of the sort developed in Strand 1 of COMIC (and also in Strands 3 and 4) may often be assessed in terms of their contributions to technologies of accountability and inter-organizational ordering.

The Purposes of Models

Much of the previous paragraph emphasizes the dependence of our arguments upon the current case. While we conjecture that the concepts we have offered (technologies of accountability and inter-organizational ordering, workflow from within and without) may have utility beyond it, we are not insisting that issues to do with the visible documentation of work are always the most pre-eminent and that all models of work should be appraised in this regard in all cases. The case of PF2 at EP is a perspicuous example of these phenomena but one should not assume that they recur in all organizational contexts. In this respect, we are not offering any kind of general 'theory of technology' in our use of Suchman's (1994) phrase 'technologies of accountability' - still less any a priori ground for a morally or politically derived critique of technology on its basis[1].

Indeed, for one thing, PF2 itself in a different organizational setting might be worked with in very different ways. It has been remarked to us by staff at EP that perhaps PF2 is most appropriate for very large scale traditional print work where, for example, the overhead of using the system would be small in comparison with the length of jobs. In such a setting PF2's workflow from without might be in lesser tension with the indigenous methods for organising from within. For another thing, it would be quite possible to use PF2 in a variety of different ways to fulfil a variety of different purposes in organizational contexts which were not as subject to the demands to work in ways so that their work can be made inspectable as is the case at EP. Indeed, this is encouraged by PF2's manufacturers promoting it as a multi-purpose system: stock control, cross-site monitoring, quality assurance et cetera are all supposed to be supported by it. In short, while 'workflow from within/without' and the other notions we have offered give us analytic leverage on the PF2/EP case, it should not be routinely assumed that all cases can be perspicuously analysed in this way.

There is a more positive and, for the present chapter, more instructive way of making this point and this is to draw attention to the purpose-relevant nature of any assessment, evaluation or appraisal of a workflow (or other) model. We have suggested that PF2's serial model of processes needs to be understood in relation to the *purpose* of conducting one's work in such a way that it can be documented and shown to be in line with stipulated inter-organizational relations. In most actual cases, such understandings will be related to a panoply of complexly interacting purposes. Again, this is the case at EP. The purpose of making one's work inspectable has to be reckoned on in relation to other management purposes (maintaining high quality product, keeping the shopfloor workforce happy etc.). It is *contingently* the case at EP that accurate and justified invoicing, demonstrations of levels of service and so forth are the pre-eminent purposes requiring satisfaction, and for the moment subordinating many others, if EP's relationship with The Department is to continue in its current form.

Models as Resources for Action (Revisited)

In COMIC, much of our understanding of the status of models of cooperative work and its organizational context can be condensed into the slogan *models are resources for action*. As such this slogan has often been in implicit or explicit contrast with the more mainstream (at least in the social, cognitive and computing sciences) *models are representations of action*. However, the study reported in this chapter highlights the importance of studying models and resources in relation to

---

1 On this last point, we feel we differ with Suchman. Suchman uses 'technologies of accountability' not merely to gain analytic purchase on a particular set of socio-technical phenomena (as we have done) but also as part of a politicised critique of certain forms of technology (as we have not).

purposes. To give a crude example, a hammer may be a resource for action if one's purpose is to strike a nail but it is somewhat resource-limited if one's purpose is to write a letter. Similarly, it is only minimally informative to note that models (of the sort embodied in CSCW systems) can be resources for action without specifying what kinds of action one is engaging in and under the aegis of what kinds of purposes.

However, none of these issues can be dealt with abstractly. One cannot, we insist, understand the status of models of work and organizational context as if the whole range of human purposes could be reduced to a few simple classes open to theoretical deduction[2]. For us, the fact that some abstraction of 'purpose-in-general' does not exist (cf. Garfinkel's, 1967, remarks on 'context-in-general'), compels us to reveal the different ways in which models of work, interaction and organization come to have whatever significance that they do in their specificity through empirical case study. Unfortunately, in CSCW, studies of models (and formalisms, representations etc.) from *this* perspective remain somewhat rare. Much of the CSCW literature has been devoted to either technical issues in model development, or empirical studies documenting the 'gap' between work and models thereof, or conceptual critiques denouncing much of the whole enterprise. In COMIC, and with the current study, we hope to have started to address this shortcoming.

Design Requirements for Cooperative Technology

It is not within the available scope of this chapter to offer detailed recommendations for CSCW technology as a result of our fieldwork. In fact, the case of PF2 at EP has caused us to reflect upon the larger issue of the very relationship between fieldwork findings and CSCW systems, and just how the former might influence requirements for the latter.

On the one hand, the methods we have uncovered by means of which print workers organize the flow of work from within can be taken to point to domains for application support. We would not be the first to commend that CSCW systems offer support for awareness and mutual monitoring or at least do not contradict members' own methods (cf. Heath and Luff, 1991; and the work supporting 'awareness' in Strand 4 which has been motivated by such considerations). Equally, as a workflow system, PF2 could be criticized for not allowing flexible mappings from processes to operators, for not specifically supporting 'run-time' re-allocations, for not recognising ad hoc collaborative arrangements, and for adding to the work that people have to do (cf. Abbott and Sarin, 1994; and the specific

---

2 This does not mean that an abstraction or simplification of this sort might not be useful in attempting to build a workflow system or a system supporting an awareness of organizational context as in the development work within Strand 1 of COMIC. For the purposes of technical development, such simplifications may be warranted and even necessary. For the purposes of gaining an adequate social scientific understanding the purpose-saturated status of models within some practical activity, we argue that such simplifications are rarely justified. (The reader should note that no overly clever self-reflexive point is being made here. We are merely being even-handed and self consistent, while not assuming that desiderata for social scintific understandings of systems crudely translate into technical requirements.)

means which have been introduced to heighten flexibility in the workflow systems presented in this deliverable). These are all familiar emphases within CSCW which our study also underlines.

However, we take it as a more challenging and urgent matter that CSCW research consider the implications for system requirements of understanding workflow technologies as *technologies for (inter-organizational) accountability*. This opens up a whole new set of issues for CSCW requirements. First of all, we feel that CSCW research must be more attentive to the formal (in the sense of 'for administrative and managerial purposes') problems that organizations face and often impact upon not only their technology policies but also the details of usage. Accordingly, we are worried about the equation of CSCW with *informal, non-structural interaction* that some researchers make. It is not that we advocate traditional 'structural' notions of the organization. Far from it (see Deliverables D1.1 and D1.2). Rather, we wish to draw attention to the multiple considerations which impinge upon the acceptability of technology in actual contexts - considerations which often require very difficult trade-offs. If, for example, there are good organizational reasons for accounting for the work in new and more detailed ways, how are these to be balanced up against the requirements of smooth workflow on the shopfloor or in the office? Indeed, in the current case, one might even argue (after all!) that a workflow system like PRINTFLOW PF2 *is* a reasonable solution, *provided* an organization anticipates the extra work and reckons on it as a cost in bidding for new business, *provided* those offering work for tender do not incorporate demands which might rebound on them, *provided* tenderers do not make similarly unrealistic promises, *provided* an appreciation of how workflow is organized from within and can be disrupted from without is maintained by all parties and so forth.

If CSCW research is to learn one thing from settings like the one we have studied, it is that a naive view of cooperative work and its support has no place on the shopfloor. Organizationally acceptable technology is achieved less through the pursuit of ideals than by ensuring that the list of *provisos* is tolerably short.

Acknowledgements

We are grateful to all personnel at 'Establishment Printers' for their willingness to have us hang around and ask obvious questions. James Pycock and Tom Rodden have given useful suggestions for improving this paper. John Bowers and Wes Sharrock acknowledge help and support from Rank Xerox Research Centre Cambridge Laboratory.

References

Abbott, K. & Sarin, S. (1994). Experiences with workflow management: Issues for the next generation. In *Proceedings of CSCW '94, Chapel Hill, USA*. New York: ACM.

Agostini, A., De Michelis, G., Grasso, M. & Patriarca, S. (1993). Reengineering a business process with an innovative workflow management system: A case study. In *COOCS '93, Proceedings of ACM-SIGOIS Conference on Organizational Computing Systems*. New York: ACM.

Bowers, J. & Churcher, J. (1988). Local and global structuring of computer mediated communication: Developing linguistic perspectives on CSCW in Cosmos. In *CSCW '88: Proceedings of the second conference on Computer Supported Cooperative Work*. New York: Association of Computing Machinery.

Bowers, J. (1992). The politics of formalism. In Lea, M. (ed.) *Contexts of Computer Mediated Communication*. Brighton: Harvester.

Bowers, J. (1994). The work to make a network work: Studying CSCW in action. In *Proceedings of CSCW '94, Chapel Hill, USA*. New York: ACM.

Carley, K. & Prietula, M. (1994). *Computational organization theory*. Hillsdale: LEA.

Coulter, J. (1983). *Rethinking cognitive theory*. London: Macmillan.

De Michelis, G. & Grasso, M. (1992). Situating conversations within the language/action perspective: The Milan Conversation Model. In *Proceedings of CSCW '94, Chapel Hill, USA*. New York: ACM.

Doran, J., Palmer, M., Gilbert, G. N. and Mellars, P. (1994). The EOS project: Modelling Upper Palaeolithic social change. In G. N. Gilbert & J. Doran (eds.) *Simulating societies: The computer simulation of social phenomena*. London: UCL.

Dreyfus, H. (1979). *What computers can't do: A critique of artificial reason*. New York: Harper & Row.

Garfinkel, H. (1967). *Studies in ethnomethodology*. Englewood-Cliffs: Prentice Hall.

Garfinkel, H. & Sacks, H. (1970). On formal structures of practical action. In J. McKinney & E. Tiryakin (eds.) *Theoretical Sociology*. New York: Appleton-Century-Crofts. pp.337-366.

Gilbert, G. N. & Doran, J. (1994). *Simulating societies: The computer simulation of social phenomena*. London: UCL.

Hammer, M. & Champy, J. (1993). *Reengineering the corporation: A manifesto for business revolution*. London: Nicholas Brealey Publishing Ltd.

Heath, C. & Luff, P. (1992). Collaboration and control: Crisis management and multimedia technology in London Underground Line Control Rooms. *Computer Supported Cooperative Work*, 1 (1/2), 69-94.

Hughes, J., Randall, D. and Shapiro, D. (1992). Faltering from ethnography to design. In *Proceedings of CSCW '92, Toronto, Canada*. New York: ACM.

Krackhardt, D. (1994). Graph theoretical dimensions of informal organizations. In Carley, K. & Prietula, M. (eds). *Computational organization theory*. Hillsdale: LEA.

Medina-Mora, R., Winograd, T., Flores, R. and Flores, F. (1992). The action workflow approach to workflow management. In *Proceedings of CSCW '92, Toronto, Canada*. New York: ACM.

Orlikowski, W. (1992). Learning from Notes: Organizational issues in groupware implementation. In *Proceedings of CSCW '92, Toronto, Canada*. New York: ACM.

Reynolds, R. (1994). Learning to cooperate using cultural algorithms. In G. N. Gilbert & J. Doran (eds.) *Simulating societies: The computer simulation of social phenomena*. London: UCL.

Robinson, M. & Bannon, L. (1991). Questioning representations. In *Proceedings of ECSCW'91, Amsterdam, The Netherlands*. Dordrecht: Kluwer.

Schmidt, K. (1991). CSCW or riding a tiger. In *Proceedings of ECSCW'91, Amsterdam, The Netherlands*. Dordrecht: Kluwer.

Shi, P., Luh, P. & Kleinman, D. (1994). Team coordination under individual and team goals. In Carley, K. & Prietula, M. (eds). *Computational organization theory*. Hillsdale: LEA.

Suchman, L. (1987). *Plans and situated action: The problem of human machine communication*. Cambridge: CUP.

Suchman, L. (1994). Do categories have politics? The language/action perspective reconsidered. *Computer Supported Cooperative Work*, 2 (3), 177-190.

Winograd, T. (1994). Categories, disciplines and social coordination. *Computer Supported Cooperative Work*, 2 (3), 191-197.

Winograd, T. & Flores, F. (1986) *Understanding computers and cognition: A new foundation for design*. Norwood: Ablex.

Chapter 3: Hunting the Snark? The Case for & against Organizational Memory

Liam J. Bannon
University of Limerick, Ireland

Kari Kuutti
University of Oulu, Finland

Introduction

In many disciplinary endeavours, there are certain concepts and issues that simply will not go away, despite repeated attempts to undermine them and show that they are either illogical and stem from flawed reasoning, or are too poorly defined to be of any investigative use. One such concept, that can still be found in several fields under somewhat different guises, is the subject of this chapter, namely the concept of "organizational memory". This hoary old chestnut can be found in such different fields as administrative studies, organizational theory, change management, psychology, sociology, design studies, concurrent engineering, and software engineering. Indeed, the term seems recently to have gained increasing prominence, with a number of workshops and panels devoted to it at meetings, including this present one. The fact that such a concept, even if under-defined, is appealed to across a wide range of studies is testimony to the fact that even if people cannot agree on what the term means, there must be some set of issues that can be subsumed under its umbrella that people feel are important and worth discussing.

The intent of this chapter is to once again examine the concept of "organizational memory" in an effort to disentangle some of the mixed, and occasionally - in our opinion - wrong-headed, views on the topic. Whether or not one believes that such a concept exists *per se*, or is simply another example of the "fallacy of misplaced concreteness" we hope to provide some insight into the themes underlying these discussions. In the next section we start out our hunt for the meaning of organizational memory through examining certain papers in administrative studies that utilise the concept, and then follow-up with a somewhat different perspective on the phenomenon, through investigating a number of computer-based projects that explicitly take as one goal of their activity an active support mechanism for this phenomenon. Thereafter we investigate some earlier psychological research on the concept of memory and show that the distinction between memory as a passive store and memory as a constructive act has a long history. We then attempt a re-construction of the classic notion of organizational memory based on the active aspects of remembering, rather than accessing passive memory stores. Finally, we indicate some implications of our approach for the field.

Current Work on Organizational Memory

One of the features that any student of organizational memory is likely to note is the many and varied locations where this term can be found, as it does not exclusively "belong" to any particular research area or discipline. That said, it is of course true that the bulk of the work that wrestles with the meaning and use of the term can be found in the field of organization studies. Given the shortage of space in this chapter, this is not the place to provide an exhaustive or even extensive reference to the evolving literature on the theme. Rather, we will refer to a recent major review paper on the topic (Walsh & Ungson) which collates much of the early material and provides a useful context for our subsequent remarks. Thereafter we switch our attention to the more pragmatic concerns of systems builders, and their attempts to implement systems that support some conception of organizational memory.

Organizational Analyses

In the broad organizational and administrative studies literature, one can find approaches to organizational memory that either focus on the individual cognitive capabilities of people and view organizational learning and memory as simply defining the learning or memory of individuals in the organization as organizational, present organizational memory as some cognitive property of an organizational collective entity that can be viewed as learning and memorising just like people[3] (Cook and Yanow, 1993):

> "The concept of memory is commonly understood to be analogous to a storage device where everything we perceive and experience is filed away." (Kim, 1993), p. 39.

> "Given what is known about the many factors contributing to inaccurate learning and incomplete recall and to motivational distortions in sharing information, it is not at all surprising that the human components of organization memories are less than satisfactory (...)In the future, smart indexing or artificial intelligence will facilitate retrieval of transaction information and will result in computer-resident organizational memories with certain properties, such as completeness, that are superior to the human components of organizational memories" (Huber, 1990), p.60.

---

[3]This links into the literature that refers to organizations as "giant brains".

In the latter case also the cognitive psychology conception about human as a fallible and limited "element" in systems have been brought in with the memory idea. We will later on return to the roots of this implicit use of psychological concepts and question the validity of the endeavour.

Another notion of organizational memory in organization and administrative literature is the idea that everything in organizations contains some information and is thus comparable with a "memory":

> "Organizational memory, broadly defined, includes everything that is contained in an organization that is somehow retrievable. Thus storage files of old invoices are part of that memory. So are copies of letters, spreadsheet data stored in computers, and the latest strategic plan, as well as what is in the minds of all organizational members" (Kim, 1993).

One of the best known and widely cited conceptions of organizational memory is presented by Walsh and Ungson (Walsh and Ungson, 1991). In their broad paper Walsh and Ungson develop from the administrative science point of view a model of the structure of organizational memory, that synthesises a large part of the previously presented conceptions. The fundamental component of this model consists of five "retention bins", around which both acquisition and retention take place. The "bins" identified are the following ones (pp. 63-66):

1. *Individuals*. Individuals store information about their organizations in their own capacity to articulate and remember experience, and they keep records and files as memory aids.
2. *Culture*. Culture embodies past experience that can be useful in dealing with the future. The cultural information is seen to be stored in different ways, as in language, in shared frameworks, in symbols or in stories.
3. *Transformations*. Transformations embody the logic that guides the transformation of an input (e.g. raw material, a new recruit, an insurance claim) into an output (correspondingly, e.g. finished product, a company veteran, an insurance payment).
4. *Structures*. Different roles within an organizational structure provide a repository in which organizational information can be stored.
5. *Ecology*. The actual physical arrangement of a workplace embodies also information about the organization that can be potentially decoded.

The conceptual framework that is proposed in Walsh & Ungson is comprehensive, but it suffers from an attempt to include everything, so that one is left wondering exactly what, within organizations, is not a part of organizational memory? There is also a distinct bias towards a storage model of “memory” in the paper, despite occasional references to active features of using memories. This is another point that we will return to later in the chapter. We would agree with Cook that neither of these two alternatives seems satisfactory. The former tends to eschew any analysis of the real organizational context of the activity, whereas the latter lends itself to an anthropomorhism, which is distinctly problematic and unhelpful, leading us to search for the location of engrams in the basements of an organization.

Computer-based Systems to Support Organizational Memory

While it is important not to reify this concept of “organizational memory”, and as we have seen the term has been used in a very wide and to us somewhat confusing way in much of the organizational theory literature, the label still serves as a useful heuristic to describe a set of concerns about how information is collated, stored, accessed, accreted, updated, and used in organizations. We now turn to a short examination of a quite different body of literature within the technical domain, where another perspective on this concept can be found, embodied in particular prototypes of software systems.

Within the computing community, a number of pilot and even some commercial systems have been developed to provide some form of what has been termed organizational memory (the pioneering work of Engelbart and the NLS team on the NLS Journal system and “community handbook” (Engelbart, 1963), (Engelbart and Lehman 1988)). Also, there is work in engineering design concerning shared memory (Konda et al., 1992), while others have tried to develop systems to support the software development process through maintaining a design rationale (Conklin and Begemen, 1988) or a design knowledge base (Terveen et al., 1993) or more general systems support (Ackerman, 1994b) and organizational support (Fuchs and Prinz, 1993). Let us now briefly characterise some of the contributions at the level of systems before we go further into our analysis of the concept.

Doug Engelbart’s Vision

One of the few people who foresaw the revolutionary potential of the computer as a medium for improving idea development and group and organizational communication was Doug Engelbart, who conceived a project entitled "Augmenting the Human Intellect" at Stanford Research Institute in the early sixties (Engelbart, 1963). This work of Doug Engelbart and his group has had, and indeed is still having, a profound impact on the development of interactive computing and our interfaces to computers (Engelbart, 1988), (Bannon, 1989). Engelbart’s vision encompassed a new kind of computerized working environment in which the emphasis was on how people could achieve significant gains in productivity as a result of the computerized support made available to them. Integral to Engelbart's scheme was the provision of computerized support to enhance communication and collaboration between people. As well as providing electronic mail facilities on his system, users could link their screens together and thus work in a shared space mode, often with a telephone connection as well, so people could discuss and change the joint document they were viewing. With regard to the concept of an organizational or community memory, the system provided a Journal facility for archiving messages and reports to serve this function. Items in this record could be directly referenced in messages, and the receiver could get access directly to the referenced document if required. As well as simple archiving features such as the NLS Journal, he had an explicit design intention to provide in the planned ARPANET Network Information Center (NIC) a community support centre that would support the community by integrating and facilitating dialogue and evolving what he refers to as a community handbook, which is a "system designed to support collaboration in a community of knowledge workers." This would allow for the creation, modification, transmission etc. of messages supporting cross-referencing, cataloguing and indexing and "should also support managing externally generated items." In a more recent exposition of what he had in mind, even though this aspect never really came to fruition, Engelbart & Lehman discuss the vision of this dynamic database or superdocument.

> " Tools for the responsive development and evolution of such a superdocument by many (distributed) individuals within a discipline-or project-oriented community could lead to the maintenance of a 'community handbook,' a uniform, complete, consistent, up-to-date integration of the special knowledge representing the current status of the community. The handbook would include principles, working hypotheses, practices, glossaries of special terms, standards, goals, goal status, supportive arguments, techniques, observations, how-to-do-it items, and so forth. An active community would be constantly involved in dialogue concerning the contents of its handbook. Constant updating would provide a 'certified community position structure' about which the real evolutionary work would swarm." (Engelbart and Lehman 1988)

As noted elsewhere (Schmidt and Bannon, 1992) the notion of "a uniform, complete, consistent, up-to-date integration" of the community knowledge is hardly realistic. Interpretative work remains to be done by the actors accessing the community handbook. It could indeed be a valuable *resource* for developing what Schmidt & Bannon term a "common information space" with other actors, but due to the distributed nature of cooperative work the handbook will be necessarily incomplete and partial. However, given the time period, the ideas and implementations of Engelbart's group were quite far-sighted, and his work is still worth reading today in order to understand the breadth and depth of his vision. Only relatively recently have other researchers begun to re-investigate this work in the context of the newly emerging field that has been labelled CSCW - Computer Supported Cooperative Work.

The "Answer Garden" Project - a More Modest Proposal?

We now turn to discuss a prototype system with a focused objective of making recorded knowledge in a narrow domain retrievable for future use. The system is called "Answer Garden" and was initially developed by Mark Ackerman as an MIT dissertation project in the early 1990s, and it has developed further since then. Answer Garden is one of the few organizational memory systems that have been developed to a stage, where they are actually usable in practice. Answer Garden is a hypermedia network system that combines database-like and communication features together:

> "In the standard configuration of Answer Garden, users seek answers to commonly asked questions through a set of diagnostic questions or other information retrieval mechanisms. (...) If an answer is not found or is incomplete (or if the user becomes confused or lost) the user may ask the question through the system. Answer Garden then routes the question to an appropriate human expert. (...) The expert then answers the user via electronic mail, and if the

question is a common one, the expert can insert the question and its answer back into the database" (Ackerman, 1994b), pp. 244-245.

The first application developed by using the Answer Garden Substrate — the database and communication "engine" — was aimed to serve as an X Window help system (Ackerman, 1994b), another reported usage is to help astrophysicists cope with a multitude of different software packages through which they have to run their research data (Ackerman and Mandel, 1995). The early papers on Answer Garden had grand visions (Ackerman and Malone, 1990), but later papers are more modest, when it comes to the scope of application of the system. Ackerman has acknowledged the problems involved in interpreting preserved data (Ackerman, 1994a) due to contextual factors, and in his latest paper (Ackerman and Mandel, 1995) he explicitly advocates "memory-in-the-small" — task-based data that is so local and short-term that there should be no problems in interpreting it, as alluded to in the work described earlier.

Jeff Conklin's work on gIBIS & CM/1

The work of Jeff Conklin and others on capturing design rationale using the IBIS (Issue Based Information System) framework within a computer-based hypertext framework has been described in several papers. This viewpoint attempts to capture the existing conversations and information flows as a source of design rationale. Yakemovich & Conklin claim that:

> "The IBIS structure of Issues (which state questions or problems), Positions (which state possible resolutions of an Issue) and Arguments (which state pros and cons of Positions) is one form of the natural, intuitive structure of decisions: some choice to be made, some set of alternatives, some tradeoff analysis among the alternatives (optional), and a commitment to some resolution."

In earlier work at MCC, Conklin and colleagues developed a computer-based graphical IBIS. The intent was that this system would help groups to capture the design rationale of their projects in the course of actually making the design. Experiences of the use of this system in groups are reported in (Yakemovic and Conklin, 1990), and in (Selvin, 1994). In the former case, in a commercial software development project over an extended period, the authors claim that the method was an improvement over unstructured notes and had several beneficial side effects. CM/1 is the PC-based commercial product developed by Conklin's Corporate Memory Systems, Inc. - a spinoff company specializing in technology to provide for organizational memory and learning through "living documentation". According to CMS,

> "CMS's products for organizational learning are based on two technical insights. The first is that decisions, assumptions, and open issues (i.e. events surrounding the making of intellectual commitments) are the pivotal elements on which an organization's actions turn, that this information is never systematically recorded, and that it can be naturally and powerfully captured using a simple method. The second is that for organizational memory to be effective it must dwell within a "living document", that is, it must be embedded in the everyday tools and practices of the organization in a way that makes adding information to it and retrieving information from it easy, natural and compelling."

Selvin discusses issues concerning the facilitation of meetings with CM/1, and notes some of the problems that can occur in trying to use the system in real-time at meetings, such as the problem of classifying the rhetorical type of an utterance and placing it on the decision map (issue net) in an appropriate place quickly enough so as not to inhibit the conversations. He also notes that the "culture" of specific groups can be different, as to their acceptance of the new language and way of discussing issues that is required in using such a tool. In some cases, it has been noted that people who are supportive of the methodology will code up discussions in this formalism after the meetings, rather than having group acceptance and collective use of the tool. In terms of our immediate interest in organizational memory, while this approach is claimed to have the potential for the management of longer-term group memories, it has not yet been fully supported in the tools, to our knowledge, and to date there is little information available on subsequent re-use of this information.

Living Design Memory

A recent paper by software developers at AT&T Bell Labs provides an interesting and thoughtful discussion of issues surrounding the concept of an organizational memory as well as a description of the development and use of a prototype system to serve as a "living design memory" (Terveen et al., 1993). Like many others in the area of software development, the authors are concerned with the high cost of developing software and have developed their tool in an effort to integrate local design knowledge, rules of thumb, heuristics, lessons learned from previous designs, etc., into an evolving knowledge base that is constantly evolving through use. What is striking in the account of what they learned in the process is the fact that the relevant knowledge exists in the form of "folklore" rather than being enshrined in formal organizational procedures, and their recognition of the need to integrate their system into the everyday organizational practice of the community if it is to serve any function: "the members of the community in which a system is to be deployed must *own* the system" (original italics). Contrary to many in the field of information systems, they recognize that: "knowledge of *facts* is not enough: it also is necessary to know *how the knowledge is to be used*" (original italics). Thus early attempts providing a corporate memory involving on-line structured text files encountered problems due to the fact that the information was not organized for efficient access (the problem of indexing), there was no way to ensure compliance and no natural way to ensure the evolution of documents. Their solution was to develop a design knowledge base and a designer assistant program which interfaces between the designer and the system, giving advice which the human designer should incorporate into their design document. At design review further information produced is fed back in to the design knowledge base. The paper is very interesting because it provides an account of the iterative design of the system based on experiences of use of the prototype. At the same time, however, we should note that many users still have problems with the current system. The authors claim that this work, while related to that of Conklin described above, goes beyond capturing design rationale and does not stop at integrating a tool into design practice, as with gIBIS, but also at integrating it into existing organizational processes, modifying these processes as necessary.

Design Engineering

Konda et al. (1992) have written an interesting paper on organizational memory from the viewpoint of engineering design, where they explicitly address the problem of contextuality we have pointed out above. In the paper, they trace the variety and development of different design theories and come to the conclusion that universal design methods have a multitude of problems and that in order to use them they should be contextually evaluated using collected historical experiences. To facilitate this process, they suggest the necessity of a "shared memory". The shared memory concept by Konda et al. can be divided in two forms: vertical and horizontal. Vertical shared memory is the collected corpus of knowledge within one professional group or subdiscipline within such group. This knowledge is more or less universal, collected in textbooks and advanced by research. Horizontal shared memory is a corpus of knowledge with a consensus and meaning shared by different professional groups and disciplines participating a particular design project. Konda et al. insist that some form of a shared horizontal memory is a necessity for any design project. Thus they not only recognize the importance of the maintenance of the contextuality of information, they make it a prerequisite. We will return to some of their concerns in the final section of this chapter.

Summary

What we have seen from the brief accounts of the systems described above are wide disparities in the conceptual frameworks employed, and the empirical evidence in support of the systems developed, yet undoubtedly, these researchers have tapped a rich vein, as they all are of the opinion that some form of shared memory is of importance to organizational development, even if there is still profound disagreement and confusion about exactly what kind of computer support might be possible to enhance this process. While a large part of the work in say, CSCW, especially in the area of CSCW software development, has to date focused on synchronous interactions, it is likely that in the long term, support for various forms of information gathering and dissemination activities will come to be seen as having a much greater impact on organizational functioning. For example, within the Computer Supported Cooperative Work (CSCW) community, note that both ECSCW'91 and CSCW'94 had panels on the concept of organizational memory, both of which, in the opinions of most of the audience, generated more questions than answers. Now that we have examined a variety of approaches to understanding and implementing the idea of organizational memory, it is time to return to a more fundamental re-examination of the metaphors of memory that are implicit in these perspectives.

The Concept of Memory Re-visited

Remembering is not the re-excitation of innumerable fixed, lifeless and fragmentary traces. It is an imaginative reconstruction, or construction, built out of the relation of our attitude towards a whole active mass of organised past reactions or experience, and to a little outstanding detail which commonly appears in image or in language form. It is thus hardly ever really exact, even in the most rudimentary form of rote recapitulation, and it is not at all important that it should be so. Bartlett, 1932

While "memory "is one of the central concepts that has interested psychology since its foundation as an area of academic study, and even well before (cf. the Greek work on mnemonist's strategies, and the analysis of the Method of Loci for memorization), over the past 30 years much psychological theorizing has been influenced by work in computer science, in particular artificial intelligence, due to an interest in possible mechanisms underlying human cognitive abilities. It is not the place here to critique this particular turn of events, but it is important that its formative influence on much psychological theorising be understood as it had serious implications for the way in which conceptions of human memory became intertwined with models of computer storage. Thus, there developed a very direct and concrete linkage between the human act of remembering and some function retrieving information from a computer store. Note that no longer is the computer simply a metaphor for human cognition but rather, "computation is cognition" (Pylyshyn, 1984). The result of this takeover has been the relative casting out of an alternative conception of memory - that also has had a long lineage from Ancient Times - that stresses the active act of remembering over the notion of some form of simple table lookup. Indeed, one can see these two contrasting perspectives from the early days of psychology proper as well, with the today lesser known views of Franz Brentano and others concerning "act" psychology being defeated by the empiricist associationists. However, the concerns underlying the alternative view have never been completely discarded, and can be seen in what is unquestionably one of the landmark books in psychology by the eminent British psychologist, Sir Frederic C. Bartlett entitled "Remembering" (note: not Memory!) back in 1932. For many people who may have despaired of the meagre results and methodological nitpickings that characterise much of the dust-bowl empiricist behaviourist psychological work from the 20's to the 60's, this book will be a relief. It is full of insightful observations, clever experimentation, and thoughtful conceptualizations. Of interest here is its repeated emphasis on the view of human memory as anything but a passive store, but of remembering as a constructive act - "remembering appears to be far more decisively an affair of construction than one of mere reproduction. ...condensation, elaboration and invention are common features of ordinary remembering" (Bartlett 1932, p. 205).

During the 1970s and 80s the information-processing perspective in cognitive psychology was so dominant that its metaphors became a part of everyday talk and for a layman — including non-psychological researchers — it was synonymous with the whole of psychology. Thus Bartlett's work was not as influential as it should have been during this time. We do find renewed interest in the late seventies when aspects of the computer model were being called into question, and more emphasis was being given to ecological factors in human cognition, which did not accord well with simple computational accounts of phenomena (cf. Neisser, 1982). It is as important to remember that information processing psychology is even in the domain of cognition but one among the many traditions of psychology. Some of the other traditions have a distinctively different view on cognition and memory and it might be worthwhile to recall some of them because we believe that they offer a more realistic starting point for the discussion about memory and remembering. One such psychological tradition that also shares Bartlett's concerns with the active nature of human memory processes that emphasises remembering as purposeful action in some definite context is Russian cultural-historical psychology, founded by L. S. Vygotsky in the 1920s. Cultural historical psychologists see remembering as processes of structuring and storing past experience to make possible its use in activities. It is a purposeful action relying on the use of socially developed signs and depending on the goals and motives of the activity within which it takes place. One of the best-known memory researchers within the tradition is a contemporary of Bartlett, P. I. Zinchenko:

> "Within this framework, memory processes can be viewed neither as a mechanical coupling, as a connection, of subjective images and experiences nor as external relations. To treat them as the function of some metaphysical capacity for memory, a capacity to preserve and reproduce impressions, is also unacceptable. Nor can they be viewed as a metaphysical capacity of the brain, of the brain conceptualized outside the actual process of the subject's life. Memory processes must be understood as processes that constitute the content of a specific action. They must be understood as remembering or recollection responsive to and functioning in a particular task." (Zinchenko, 1983), p.76.

Each action of memorizing or storing information and each action of recalling and remembering take place in the context of an activity. If storing context and recalling context are the same activity, the interpretation of the material may not be problematic. But if remembering takes place in a different activity where material has been stored, the material will be reinterpreted with respect to the new object of activity, and there is no automatic guarantee that the material is relevant anymore in the same way than it was in the context of storing it. We believe that this problem of contextuality has been somewhat neglected in the studies of reuse, design rational, etc. mentioned earlier and further elaborations are necessary.

The importance of this work for our discussion here on the much broader concept of organizational memory is that in the vast majority of cases, underlying any mention of the term memory is a view of memory as some passive register of experience. Yet we now know that such views are certainly not appropriate to understand human memory. This does not of course therefore imply that providing some register of events or some form of storage is inappropriate in an organization, as of course we are required for legal reasons alone to maintain such records, but it does become important in situations where people are designing systems that are supposed somehow to allow people in organizations to store and later retrieve accounts of experiences which can hopefully be shared throughout the organization. So, what are the consequences of taking such an approach to organizational memory? In the next and final Section, we begin the process of re-constructing this concept according to this alternative perspective, and hint at possible questions that are raised, and issues that need to be explored more fully.

Implications / Consequences of our position

□Organizational knowing is a process that depends on uncertainty, improvisation, dialogue and conflict.□ □Blackler, 1993□

The purpose of this chapter has been to survey a number of quite disparate activities in a variety of fields concerned with the theme of organizational memory and learning. We have shown how much of the work in the organizational field has been based on an implicit, if not explicit, view of organizational memory as akin to human memory. Models of human memory that tend to be discussed emphasize human memory as the storing of experience, so memory is viewed as a storage bin. How to "capture" information in the organization, and then re-circulate it then becomes simply a matter of developing suitable hypertext and electronic communication systems to help in the "input" and "output" of the engrams, traces, or information nuggets that exist in the organization. It is in a sense a great irony that in much of this work we have a circular set of concepts and definitions - computer systems are used by information-processing psychologists to develop theories of human memory, these models of human memory are in turn influencing organization theorists in their views of organizational memory, and in turn are the basis for computer systems!

We have examined briefly some computer systems that have been developed to support aspects of organizational memory, and noted some of their features, positive and negative. While some of these systems have been developed with a limited pragmatic purpose in mind, in the majority of cases there is still an implicit perspective of "capture" of relevant information. What we have attempted to show in the middle section of this chapter is that there is an alternative conceptualization of memory which has a long tradition that focuses on the active constructive aspects of remembering. In our view, this perspective is not only of importance for psychological or sociological theory, but has implications for the construction of computer "support" systems for any such processes. For example, a number of information systems projects that attempt to capture all the activities of groups within an organization would appear to be going up a blind alley, as such data capture is unlikely to be able to be interpreted and re-worked to be useful for a later situation. Time passes, and both the people, settings and context in which the original "information" was produced change also. Thus the likelihood of being able to characterise what kinds of information in an organization are potentially significant and worth keeping is an impossible task, as we must take into account the fact that people are actively making sense of the information presented, either intra- or inter-subjectively.

How do we "capture" these meanings that are required in order to make sense of any situation or fact? Our very concepts for discussing such issues are not well developed, although recently there has been work from a variety of quite different sources which at least acknowledges the problem, and offers some suggestions as to what direction we might head. For example, in an ambitious and important programme of work briefly discussed earlier (Konda et al., 1992), a group at CMU in Engineering Design are involved in the building of a shared memory, but in contradistinction to much of the work in the area, they have taken on board some of the concerns expressed here. They are aware that collaboration does not simply consist of a transfer of information between parties but that for any sort of shared memory to be developed there must be shared meanings: “one cannot have a meaningful shared memory without shared meaning, since a memory that is neither accessible nor understandable can hardly be called sharable“ (Konda et al., 1992). This view is strikingly reminiscent of comments made in another paper, concerning the notion of a “shared information space“ (Bannon and Schmidt, 1991) or more recently, a “common information space” (Schmidt and Bannon, 1992): “A common information space encompasses the artifacts that are accessible to a cooperative ensemble *as well as* the meaning attributed to these artifacts by the actors.” They elaborate ”Objects must thus be interpreted and assigned meaning, meanings that are achieved by specific actors on specific occasions of use. Computer support for this aspect of cooperative work raises a host of interesting and difficult issues that have not been fully addressed within the field to date.”

The implications of these views for building corporate repositories of information is only beginning to be addressed. In both cases, the problem resides in the fact that information does not simply exist “out there”, but is produced by specific people in specific contexts for specific purposes. While this does not imply that it is bound solely to that whole context, it does mean that one cannot in any straightforward way extract and abstract from this web of signification items of “information” which can be stored in some central resource for later use without having some conception of this whole “context” question. What is good information changes depending on the time, the originator, the context, etc....and without these cues, the relevance of items of “information” becomes deeply problematic. The views of the CMU group, Bannon & Schmidt, and the authors of this paper is that no universal language will be possible for encoding information, nor is there any algorithm to determine “relevance”. Information is always produced in a context, and must be re-interpreted in other contexts. Understandings, either between people or between artifacts or information and people, are achieved, not given. Neither human remembering, nor human interaction simply occurs, but it is an outcome that is dependent on the interplay of many factors[4].

What is surprising is that there has been little focus among the various disciplinary groups concerned with organizational memory on the details of how organizations actually develop and use organizational memories - the ways in which procedures embed knowledge, the possibilities for changing organizational routines

---

4 In discussing human remembering, Bartlett refers to this set of factors as a “schema” although he was well aware of the possible misuses of this term, and certainly his notion is far removed from later AI attempts to reify this concept (Minsky, Schank).

as a result of organizational learning, the ways in which artefacts and their uses can inculcate a particular way of doing things throughout the organization, the care and evolution of corporate information repositories, the role of gossip and the grapevine in contributing to organizational memory, etc. The material that does bear on such issues is often developed by people from outside this community. For example, the work of JoAnne Yates on the history of managerial control and communication mechanisms in American organizations provides a rich historical analysis of material of relevance here. Likewise, from a cultural-historical activity perspective, the work of Engeström and his colleagues, e.g. (Engeström et al., 1990) is concerned with the historical analysis of work activities as a part and parcel of their developmental work research tradition. Also, there are numerous ethnographic studies of work that provide important insights into how people use records, documents and artefacts of all kinds to accomplish their work activities, and engender shared ways of viewing the world within specific communities. (see e.g. (Suchman, 1987), (Heath and Luff, 1992), (Hughes et al., 1993), etc.). Perhaps what is required within the management information systems community is more involvement in analysing the ways in which organizational memory - in whatever form it is conceptualised - is built and used in real organizations than in developing additional corollaries or hypotheses about the nature of the beast[5]!

The role of "war stories" that are swapped around among various groups, detailing interesting, difficult problems with equipment, and their resolution, is relevant here. As Bannon (1991) notes: "These stories not only impart information, they also provide a context for use of the information, and they also serve as a way of bonding the group together. They are vehicles for group cohesiveness and identity, and as such cannot be replaced with simple factual information about the original problem that is the basis of the story. Can such stories be put into a community information base without losing their dual function as both information bearing and social bonding entities? We must admit we cannot answer that question at this stage. (See (Orr, 1986), for some discussion of this topic of war stories and their role in organizations). What are the pre-conditions for having people commit to contributing and sustaining such a system? Can the motives be completely altruistic? What are the rewards, both personal, social, organizational, for those that contribute to this information repository, either directly, or when explicitly asked? What kind of support structures, either embedded in the computer network itself, or external to it, might be of use to support this kind of cooperative learning and exchange of information? Are there software needs that can be identified that would assist in the development of such a community memory?" (Bannon, 1991).

While our emphasis in this chapter has been to bring to the attention of researchers on organizational memory a hitherto relatively neglected body of

---

[5] It is interesting to speculate about the relationship between the ideas of business process re-engineering (BPR), or at least one variant of it, that argues for doing away with traditional practices in a wholesale fashion, on the one hand, and the concern with aspects of organisational memory, the preservation of aspects of organisational tradition, on the other. Investigating this seeming paradox would take us too far afield however, but see (Bannon, 1994) and (Kuutti, Virkkunen & Young 1995) for some further comments.

psychological literature that provides a re-framing of the nature of human memory and of the “memory” concept per se, it is also important to note that within the field of sociology there is also a strong body of work that emphasizes the constructive aspect of remembering as a social phenomenon rather than memory as some passive store (Hughes et al., 1995). Both sets of views reinforce the position that, at a pragmatic level, computer-based support systems for organizational memories that simple consist of some passive capturing, storage and eventual re-play of information will have very limited if any use for the practical accomplishment of activities within an organization.

In recent years, we have witnessed the development of a variety of accounts of phenomena that taken together, present a very strong case for the importance of the contingent nature of human activities, that stress the role of talk and interaction as the basis for mutual understanding and intelligibility. There is an increased interest in the role of stories and narratives as methods for encoding and disseminating information in all aspects of human life. It is not the stories per se but the discussion and debate that they stimulate that is important in developing real understanding. Wynn (1979) notes "In an office as it presently operates, the knowledge which is both means and product is dependent on interaction between people for its quality, relevance and appropriateness. These interactions are in turn dependent on social practices" (Wynn, 1979) pg. 165. More recently, Blacker notes: "Talk about computer-mediated information and the transformation of isolated problem-solving attempts into a shared activity are crucial to the effective operation of the "informated" organization. It is only through such processes that the process of collective interpretation can be reached" (Blackler, 1994), p.12. Within the CSCW community, the work of Julian Orr on story-telling as an important practice in learning on-the-job has attracted attention: "Diagnosis is observed to have a strong narrative component in the integration and assessment of known facts; the technicians tell themselves what they know about the machine. This narration prepares them to tell others of their experience, either in asking for help or telling of a new problem, and stories of interesting problems circulate quickly through the community. These stories inform the community; they also demonstrate and celebrate the competent practice in maintenance of the service situation which is the basis of the community." (Orr, 1992), p.6. As Brown & Duguid note:

> "In some form or another the stories that support learning-in-working and innovation should be allowed to circulate. The technological potential to support this distribution — e-mail, bulletin boards, and other devices that are capable of supporting narrative exchanges — is available. But narratives, as we have argued, are embedded in the social system in which they arise and are used. They cannot simply be uprooted and repackaged for circulation without becoming prey to exactly those problems that beset the old abstracted canonical accounts." (Brown and Duguid, 1991), p. 54.

In contradistinction to the explicit socially sanctioned role of story-telling, we also see an emphasis on the importance of more serendipitous talk in work settings:

> "....important function served by serendipitous talk about work is its importance in constructing and maintaining an up to date "intelligence" concerning the surrent activities of

the team. This working "intelligence" or "memory" can be seen to be collectively constituted in the team's conversations." (Middleton, 1988), p. 14.

Perhaps this interest in active forms of remembering can be connected with even broader vistas in our intellectual culture, as suggested by philosopher Stephen Toulmin:

"All the "changes of mind" that were characteristic of the 17th century's turn from humanism to rationalism are, as a result, being reversed. The "modern" focus on the written, the universal, the general, and the timeless — which monopolized the work of most philosophers after 1630 — is being broadened to include once again the oral, the particular, the local, and the timely." (Toulmin, 1990), p. 186.

References

Ackerman, M. (1994a). Definitional and Contextual Issues in Organizational and Group Memories. *27th Hawaii International Conference on System Sciences (HICSS-27), Maui, Hawaii, January 1994*. IEEE Computer Press, pp. 191-200.

Ackerman, M. S. (1994b). Augmenting the Organizational Memory: A Field Study of Answer Garden *CSCW'94, Chapel Hill, NC, USA, October 1994*. ACM Press, pp. 243-252.

Ackerman, M.S. and T.W. Malone (1990). Answer Garden: A Tool for Growing Organizational Memory. *ACM Conference on Office Information Systems*. ACM Press, pp. 31-39.

Ackerman, M. S. and E. Mandel (1995). Memory in the Small: An Application to Provide Task-Based Organizational Memory for a Scientific Community *28th Annual Hawaii International Conference on System Sciences (HICSS-28), Maui, Hawaii, 3-5 Januray 1995*. IEEE Computer Press, vol. IV, pp. 323-332.

Bannon, L. (1989): The Pioneering Work of Douglas C. Engelbart. In Z. Pylyshyn and L. Bannon (eds.): *Perspectives on the Computer Revolution*. Norwood, N.J.: Ablex, pp. 301-306.

Bannon, L. (1991): Community Technology? Issues in Computer Supported Work. *Mutual Uses of Cybernetics and Science, Special Issue of Systemica - Journal of the Dutch Systems Group*, vol. 8, part 2, pp. 23-41.

Bannon, L. (1994). Computer Supported Co-operative Work: Challenging Perspectives on Work and Technology. (Invited paper.) In W. Baets and R. Galliers (eds.): *Conference onInformation Technology and Organizational Change, Nijenrode University (The Netherlands Business School), Breuklen, The Netherlands, April, 1994*.

Bannon, L. J. and K. Schmidt (1991): CSCW: Four Characters in Search of a Context? In J. M. Bowers and S. D. Benford (eds.): *Studies in Computer Supported Cooperative Work. Theory, Practice and Design*. Amsterdam: North-Holland, pp. 3-17.

Bartlett, F.C. (1932): *Remembering*. Cambridge: Cambridge Univ. Press.

Blackler, F. (1993): Knowledge and the theory of organizations: organizations as activity systems and the reframing of management. *Journal of Management Studies*, vol. 30, no. 6, pp. 863-884.

Blackler, F. (1994). Knowledge, Knowledge Work and Organizations. An Overview and Interpretation *Workshop on European Competitiveness in a Knowledge Society, Lyon, 30/11/1994-2/12/1994*.

Brown, J. S. and P. Duguid (1991): Organizational Learning and Communities of Practice: Toward an Unified View of Working, Learning and Innovation. *Organization Science*, vol. 2, no. 1, pp. 40-57.

Conklin, J. and M.L. Begemen (1988). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *Proceedings of the CSCW''88, Portland*. ACM Press, pp. 140-152.

Cook, S.D.N. and D. Yanow (1993): Culture and Organizational Learning. *Journal of Management Inquiry*, vol. 2, no. 4, pp. 373-390.

Engelbart, D., Lehman, H. (1988a): Working Together. *Byte*, no. December, pp. 245-252.

Engelbart, D.C. (1963): A conceptual Framework for the Augmentation of Man's Intellect. In Howerton and Weeks (eds.): *Vistas in Information Handling*. Washington, D.C.: Spartan Books, pp. 1-29.

Engelbart, D.C. (1988b): The Augmented Knowledge Workshop. In A. Goldberg (ed.): *A History of Personal Workstations*. New York: ACM press.

Engeström, Y., K. Brown, R. Engeström, and K. Koistinen (1990): Organizational forgetting: an activity-theoretical perspective. In D. Middleton and D. Edwards (eds.): *Collective remembering*. London: Sage.

Fuchs, L. and W. Prinz (1993): Aspects of Organizational Context in CSCW. In L. Bannon and K. Schmidt (eds.): *Issues of supporting organizational context in CSCW systems. COMIC Deliverable 1.1*. Lancaster: Lancaster University.

Heath, Christian and Paul Luff (1992): Collaboration and Control. Crisis Management and Multimedia Technology in London Underground Control Rooms. *Computer Supported Cooperative Work (CSCW). An International Journal*, vol. 1, no. 1-2, pp. 69-94.

Huber, G. P. (1990): A Theory of the Effects of Advanced Information Technlogies on Organizational Design, Intelligenge, and Decision Making. *Academy of Management Journal*, vol. 15, no. 1, pp. 47-71.

Hughes, J., V. King, J. Mariani, T. Rodden, and M. Twidale (1993). Paperwork and its lessons for database design *12th Schärding International Workshop on Design of Computer Supported Cooperative Work and Groupware, Schärding, Austria, 1-3 June 1993.*

Hughes, J., J. O'Brien, and M. Rouncefield (1995). Organisational Memory, or, how can we sack Mavis but keep her brain? (Handout for a presentation) *COST 14 CoTech (Project 2: Common Information Spaces & Organisational Memory) Meeting, Limerick, Ireland, January 1995.*

Kim, D.H. (1993): The Link between Individual and Organizational Learning. *Sloan Management Review*, no. Fall 1993, pp. 37-50.

Konda, S., I. Monarch, P. Sargent, and E. Subrahmanian (1992): Shared Memory in Design: A Unifying Theme for Research and Practice. *Research in Engineering Design*, vol. 4, pp. 23-42.

Kuutti, K., J. Virkkunen, and K. Young (1995) *"Activity" instead of "process"? Questioning the unit of analysis in BPR.* University of Lancaster, Report No. COMIC-Oulu-1-4.

Middleton, D. (1988).Talking Work. Argument in Co-ordination, Comemoration and Improvisation in Team Work *Summer Conference on "Work and Communication", UCSD, San Diego, July 11-15, 1988.*

Neisser, U. (ed.): (1982): *Memory Observed. Remembering in Natural Contexts*. San Francisco: W. H. Freeman.

Orr, J. (1986). Narratives at Work: Story Telling as Diagnostic Activity. *Proceedings CSCW '86, Austin, Texas*, pp. 62-72.

Orr, J. E. (1992). Ethnography and Organizational Learning: In Pursuit of Learning at Work. *NATO Advanced Research Workshop "Organizational Learning and Technological Change", Siena, Italy, September 22-26, 1992.*

Pylyshyn, Z. W. (1984): *Computation and Cognition. Toward a Foundation for Cognitive Science*. Cambridge, Mass.: MIT Press.

Schmidt, K. and L. Bannon (1992): Taking CSCW Seriously. Supporting Articulation Work. *Computer Supported Cooperative Work (CSCW)*, vol. 1, no. 1-2, pp. 7-40.

Selvin, A.M. (1994): Meeting Facilitation with GDSS: Reflections on Skill Development *Proceedings of Conference on Computer-Supported Cooperative Work 1994 (CSCW'94)*. New York: ACM.

Suchman, L. (1987): *Plans and situated actions*. Cambridge: Cambridge Univ. Press.

Terveen, L.G., P.G. Selfridge, and M.D. Long (1993). From "Folklore" to "Living Design Memory" *Interchi'93, Amsterdam, April 1993*. ACM Press, pp. 15-22.

Toulmin, S. (1990): *Cosmopolis. The Hidden Agenda of Modernity*. Chicago: Univ. Chicago Press.

Walsh, J.P. and G.R. Ungson (1991): Organizational Memory. *Academy of Management Review*, vol. 16, no. 1, pp. 57-91.

Wynn, E. (1979): *Office conversation as an information medium*. Unpublished Ph.D. dissertation, University of California, Berkeley, CA,.

Yakemovic, K.C.B. and E.J. Conklin (1990): Report on a Development Project Use of an Issue-Based Information System *Proceedings of the Conference on Computer-Supported Cooperative Work 1990 CSCW'90)*. New York: ACM, pp. 105-118.

Zinchenko, P.I. (1983): The Problem of Involuntary Memory (originally published in Russian 1939). *Soviet Psychology*, vol. 22, no. 2, pp. 55-111.

Chapter 4:
The I*EARN: an Organizational Learning Network

(A different approach for the analysis of the new organizations emerging from the computer networks)

Artur Serra
Universitat Politecnica De Catalunya, Spain

Introduction.

The goals of the ethnographic work done in this research project have been the same of the rest of ethnographic work in COMIC: to analyse how a particular co-operative system works to give insights about how better CSCW systems can be designed. Yet we have adopted a different approach, instead of analyzing which kind of CSCW is required to support a given type of work, we have studied what kind of work is produced by a computer network with CSCW tools like a conferencing system.

A first result of COMIC was the production of "A Conceptual Framework for Describing Organizations" around the question: "What is an organization?". One of the COMIC conclusions is, that organizations are increasingly transformed from "bureaucratically controlled workplaces" to more "network learning organizations" (K.Kuutti &J.Virkkunen, 1994). We want to go a step further and analyse what we call "an organizational network", a computer and telecommunications network with organizational roles. I*EARN is the case study for that.

Duration.

The duration of the project has been of 2 months of virtual participant observation in this organization.

Type of Fieldwork: The Virtual Participant Observation Technique.

The traditional approach of the cultural anthropology has been looking at the organizations as cultures, i.e. communities with a shared knowledge. The cultural anthropology has stressed the knowledge aspects of a social relation, the values and common sense of the members of this organization, instead of focusing mainly on their social relations and their mechanisms of interaction.

This is the starting viewpoint of this paper. We will try to analyse an organization I*EARN as a particular network of educators sharing a common set of premises and values, that give meaning to their common practices and organizational structures. This viewpoint is particularly useful in the educational setting, where the work is based in the transmission of a common set of values and knowledge to the students. Work and knowledge are really closed in this kind of organization.

Anthropology is now increasingly working with computer cultures and we are developing new approaches in the study of the new technological phenomena. Recently the American Anthropological Association has established a group on Anthropology of Science, Technology and Computing. New research works have been produced trying to analyse the new phenomena of computing networks as new, rather sociotechnical organizations (Serra, 1992).

Anthropologists and social scientists are developing new approaches to analyse the human-computer cultures as new, rather human co-operating systems. In the traditional ethnography we use the participant observation method and the interviews with key informants, the so called qualitative methods in social research. We have been introduced in the community by a leading person in this group and invited to observe in an active way the set of projects and practices of this community.

Now working in a computer culture we need to produce some changes in these traditional ethnographic methods to adapt us to this particular kind of culture. We propose a variation called virtual participant observation. What is that? It is a technique based in putting the observer in the same "territory" as the rest of the community he is observing. In that case, as the community is a global network, the anthropologist has needed to participate *in* this network as a common electronic participant observer.

What does it mean? Take an example. I*EARN is an educational network around global projects. Their activities rely on global e-mail and newsgroups communications. A key participant invited us to participate in their internal electronic communications, particularly e-mail and electronic conferences, to understand how the student projects were built and how the life of the network went on around these projects

We applied a content analysis technique to this virtual participant observation because in this kind of organization knowledge and work are very related. Content analysis has been used by anthropologists to analyse the cultures from their own texts, verbal or written documents. All these materials are analysed as texts, and they are considered as starting points to understand the common knowledge of a community.

The Setting: I*EARN, an Organizational Learning Network.

A Theoretical Approach to the Organizational Networks.

Our theoretical framework for analysing I*EARN begins with the idea of "organizational networks". The computer networks, like Internet, APC and others, are producing two different kinds of phenomena. The first one is the changing of the work in the organizations, that have been connected to a computer network. The second one is the generation of a new kind of organization, built because of the network and on the network. We call this new kind of phenomena: the organizational networks. They are also producing a new type of work.

The COMIC Project began with an innovative effort to elaborate a "Conceptual Framework for Describing Organizations". The idea was, that understanding different theories about what organizations are and how they work, could help to the design of more useful CSCW systems.

The methodology of COMIC has been defined like this:

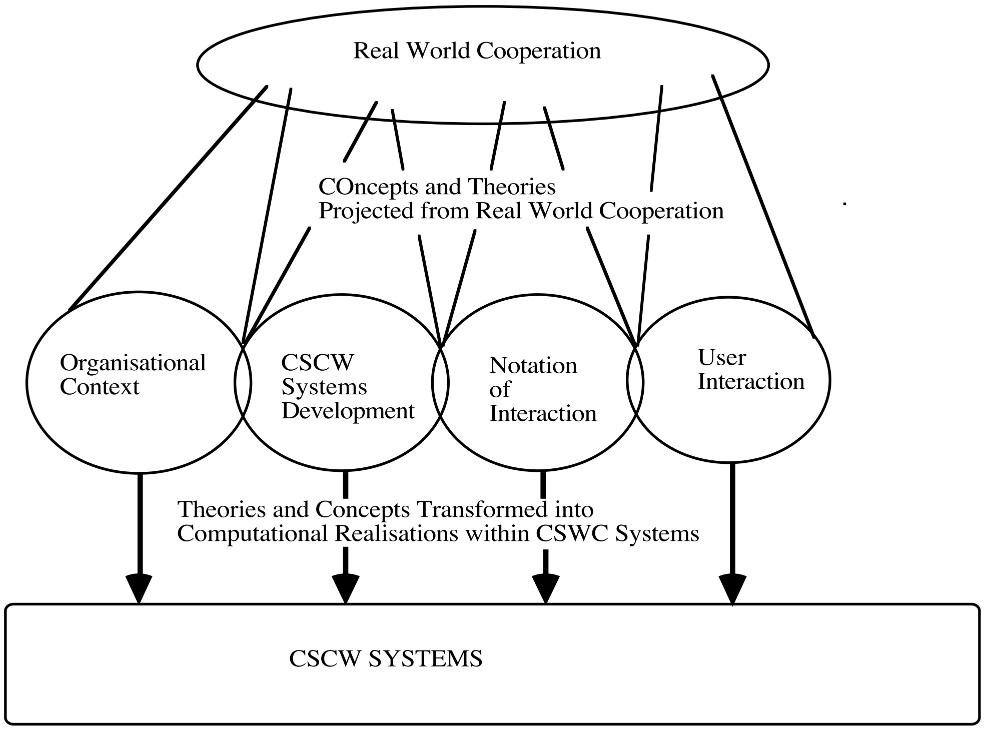

Figure 1: The COMIC methodology

This orientation starts with the study of real world organizations that are changing by the impact of information technology.

COMIC's "Conceptual Framework for Describing Organizations" began with a pluralistic viewpoint about the question: "What is an organization?" (Bowers,oct 94:1). The goal of this strand has been to produce a computational model about how organizations work to inform the design of CSCW systems. The methodology has been innovative in the CSCW field indeed: the ethnographic approach. The first field study was about "Technology in an Organization Context", in particular the introduction of a CSCW system in an organization within the U.K´s central government (J.Bowers, oct.94:13)

Studies done by social scientists about the introduction of electronic communication in organizations go back to Hiltz and Turoff(1978), Kling(1980), Bikson and Gutek (1983), Turner (1980), Zuboff (1988) and Sproull and Kiesler (1987,1991).

In their study "Connections", Sproull and Kiesler introduced the idea of the "networked organization", as a loose definition, that includes technology and people. Kuutti and Virkkunen from Oulu University (COMIC, 1994) have proposed in a very interesting paper the idea of "learning network organization" as a synthesis of three key features of the new kind of organization:

- The network forms of organizations.

- The team work and the empowerment of front line workers.
- The concept of learning organization.

This approach begins with the analysis of traditional organizations, that receive a new technology. The kind of organizations we have analysed as social scientists are almost the classical ones: nations (Hiltz &Turoff), companies and office systems (Bikson &Gutek) or traditional jobs settings (Turner, Zuboff). In brief most of the social science studies on technology have analysed mostly the "impacts of technology in society". This is still the classical question: Which impact will different technologies have on the society?.

These studies generally have shown limited results for designers of technology. They have described usually the social resistence to the introduction of computer technology. In this approach, the society is the "old" and the technology, the "new".

Take an example. Kiesler and Sproll in 1987 published "Computing and Change on Campus", an empirical study about the introduction of the Andrew system in Carnegie Mellon University. This project gathered a brilliant bunch of designers of the School of Computer Science (Newell, Borenstein, Satyanarayanan, Morris,...) with the purpose of designing a distributed network for the whole university, a distributed file system, that made this university one of the most computer intensive campuses in the United States.

Kiesler and Sproull's conclusions of this study (1987) showed a less than optimistic vision of the results of this project, and the great resistance of the faculty to adopt the premises of this so called "knowledge revolution", mainly in relation with the production of educational software.

These conclusions again showed, that if you put the society in the reactive situation, in the passive role of receiving the consequences of the technology, the society acts as a passive subject, as someone that reacts in front of a technology.

Now let us to explore another way of looking at the organizations and their relationships to technology. Suppose we analyse how the designers of technology organize themselves. Imagine that new technologies are new societies. Suppose we study not only the "impact" but the organizational generation of technology. Then the question could be, which types of new organizations are generating the computer technology. What kind of work is the responsible of the design of the new technology.

Imagine this model:

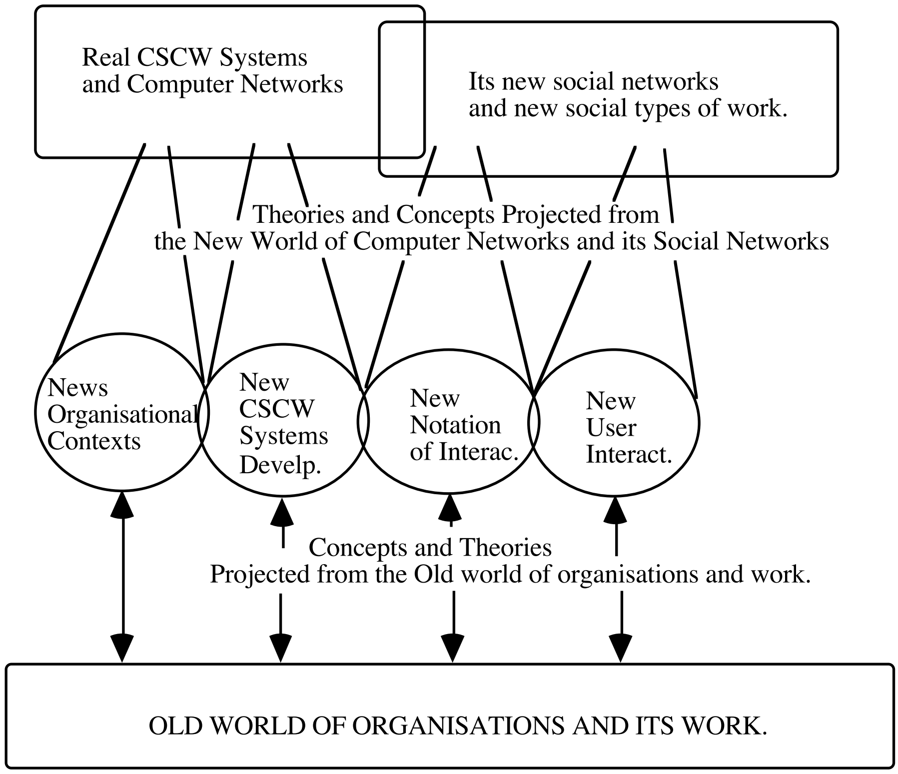

Figure 2: Organizations and technology

Take again the same example on Carnegie Mellon from the new perspective. From 1990 to 1992 we made an ethnographic study about this computer intensive campus from this different perspective (Serra,1992). What we saw was a different panorama. The computer technology in this campus was changing not only its organizations but it was produced by a new organization, the School of Computer Science. This new school generated it‘s own debate about what kind of science and technology "computer science" was, what the hierarchical status of computer science on the campus was, how the new research network and academic communities on CMU performed, what the funding sources of a research university were. In brief new technology produced and was produced by new organizations.

This implies that we have not only "the real world society" but different kinds of "real world" societies as we have different kinds of technologies. Then a key question to design CSCW systems is to begin with the study of the CSCW organizations that are designing, developping or using this kind of systems, even at a simple level.

Let us now take a look at the I*EARN case study.

I*EARN, an Organizational Network[6]

I*EARN is, in its own definition, an international, educational and resource network. Different from the traditional educational organizations that have decided to modernize themselves by the way of linking their institutions to the Internet, I*EARN is not a networked school but an educational network. Because of this difference I*EARN can launch educational projects world-wide, that connect local schools, promoting its local activities at the global arena, an unimaginable perspective for an individual primary school.

This difference separates I*EARN from the rest of the K-12 community now connecting with Internet. The K-12 community uses computer communication as a tool serving the traditional curriculum (languages, geography, history, maths,...). This corresponds to the vision of technology as something, that has an "impact" on the old organizations.

I*EARN uses the network as a new organizational way of working, that allows the introducing of new curricula's matters in the schools, the global projects. In this sense the I*EARN members use the technology as a way of new organizational activity. They organize themselves in a new way through the computer network.

This kind of work would not be possible without the use of global computer networks. In this sense I*EARN is a new computer network with new educational purposes. At the same time it is a voluntary network. Their members are not teachers forced to use the network and information technology because of public opinion or policy administrators say they have to use it. They use it because they want to and they pay for it.

I*EARN is not a network provider. It is an educational network service. For their projects they need technical connectivity through other networks. At this moment I*EARN is supported by a global computer network called APC, or Association for Progressive Communications. APC provides the technical support to I*EARN world-wide. The I*EARN computer network from the technical point of view is a collection of computers connected mostly through Internet protocols and supported by a system of computer conferences in the APC network.

---

[6] http://www.igc.apc.org:80/iearn/

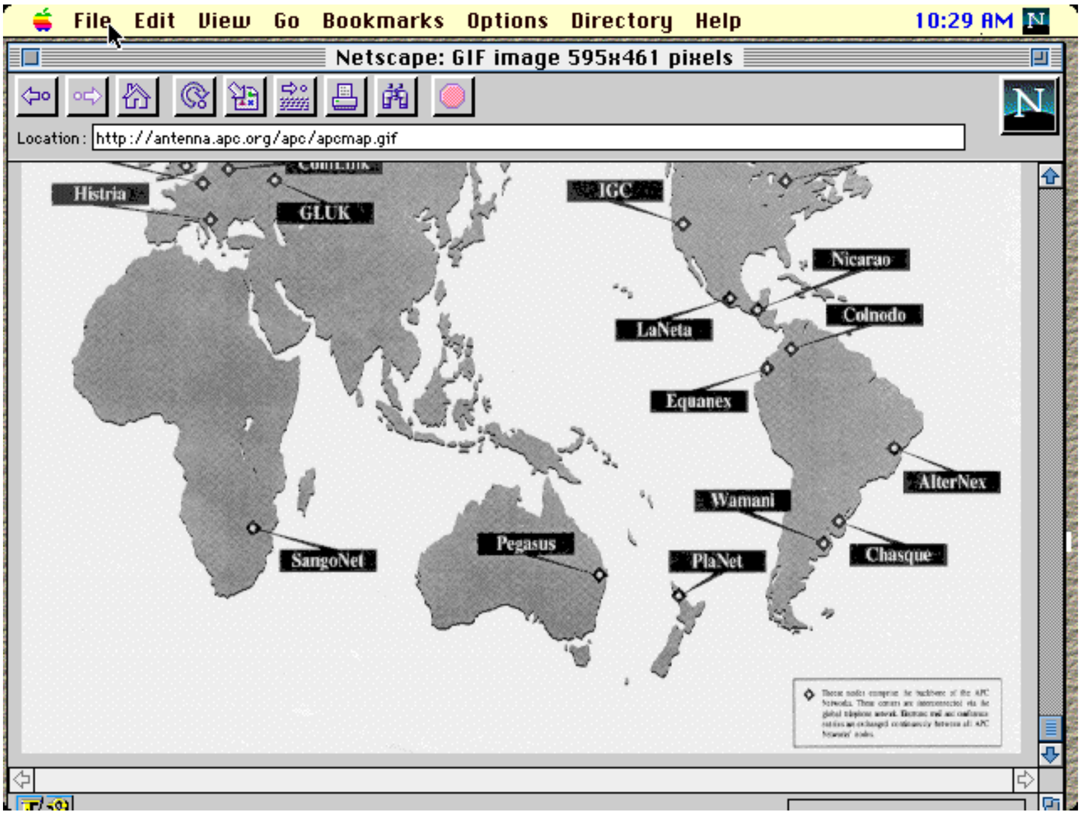

Figure 3: I*EARN in the Internet

In Spain I*EARN is supported by (http://pangea.upc.es) Pangea, a network service provider, built by ComCop (Comunicació per a la Cooperació), a non-governmental organization dedicated to give electronic facilities to NGOs.

I*EARN as a Learning Network.

I*EARN is a non-profit global network of 500 primary and secondary schools in over 20 countries. It is open to young organizations too. Its own definition of goals is the following:

> "I*EARN empowers teachers and young people (ages 6-19) to work together in different parts of the world at very low cost through a global telecommunications network. The purpose of I*EARN is to enable participants to undertake projects designed to make a meaningful difference in the health and welfare of the planet and its people. I*EARN is a non-profit organization. I*EARN is expanding to additional international sites daily and now includes over 500 schools in more than 20 countries".

By definition, they conceived I*EARN as a very simple computer supported co-operative work system ("empowers teachers and young people to work together...through a global telecommunications network"). We see the same ingenuous but useful idea, that we saw in the Andrew system at CMU: that from the point of view of users, the network is a CSCW system.

Let us underline two more aspects of their objectives:

a) "Making a meaningful difference in the health and welfare of the planet and its people".

b) "To undertake projects".

A leading participant stresses the first point: "I*EARN is for making a meaningful difference in the health and welfare of the planet, we are not interested in doing whatever educational project"[7]. Narcis Vives[8], the Spanish representative in the organization and a secondary teacher, gives an example to us: " I*EARN is not for doing a particular math exercise through the net with a limited scope and social interest. It is for projects related with global problems and questions". It is an international network dealing with global and human problems.

For our purposes we underline the character of a network-organization. It is a computer network with organizational purposes, not only technical ones.

Several members of I*EARN participate in the INET95, the international congress of Internet Society[9], with papers with a common concern: setting up a global schoolhouse. Kristin Brown, Executive Manager of the organization, presented a paper called: "Creating Global Learning Communities: I*EARN Action-Based Projects". All the I*EARN projects are selected for educational purposes. In 1995, a year declared by the UN as the Year of Tolerance, I*EARN is launching an international initiative on the same topic using an electronic conference[10].

At the same time these educational projects are intended to go beyond the class context and to have a real impact on real world situations. It seems a kind of education very closely related to social reform purposes. The teachers want, that their students not only can educate better but contribute to solving real-life problems. For Ed Gragert, acting director of the organization: "We want to see students go beyond both simply being "pen pals" and working on strictly academic work to use telecommunications in joint students projects designed to make a difference in the world as part of the educational process ".

I*EARN as a Global Network.

I*EARN was born in 1988 at the end of the Cold War as a common international project between American and Russian secondary schools. That explains its international character.

The global dimension is a characteristics of the new organizational networks, a point frequently forgotten in the CSCW community. For example, Kuutti and Virkkunen don't introduce this dimension in the three characteristics of the new network learning organization.

The industrialized bureaucratic organization was traditionally a local or a nation-wide kind of work system. Their workforce was mainly national based on national

[7]Interview, April 1995

[8]nvives@pangea.upc.es

[9]http://aleph.ac.upc.es/inet

[10]news: I*EARN.Tolerance

communications systems (railroad, electricity, telephone, newspapers, TV,...). Now the organizational networks are increasingly producing a new kind of organization, the global organization. I*EARN is one of these. APC and Internet are others.

These kinds of organizations started from the companies but from the research and educational communities. Office automation systems (Hammer, 1982; Sirbu, 1981) were conceived to capture the work process' operational details in an office setting within a very narrow scope. The global organizational network intents to support cooperative work in a large scale environment. Its goal is to embrace the whole population in a cooperative network. The office automation system's goal is to design an office completely automated without human presence.

I*EARN participates in the global organizational network model.

How does the I*EARN work: the On-line Conference Projects.

The network is based on two kinds of computing services: e-mail and electronic conferences. These two basic services are the backbone of all the network activities. Especially electronic conferences help to organize the community. In one sense, the network consists of electronic conferences.

They are divided in three kinds:

- I*EARN Conferences for projects.
- I*EARN Conferences for linking people and resources.
- I*EARN Conferences for management and training.

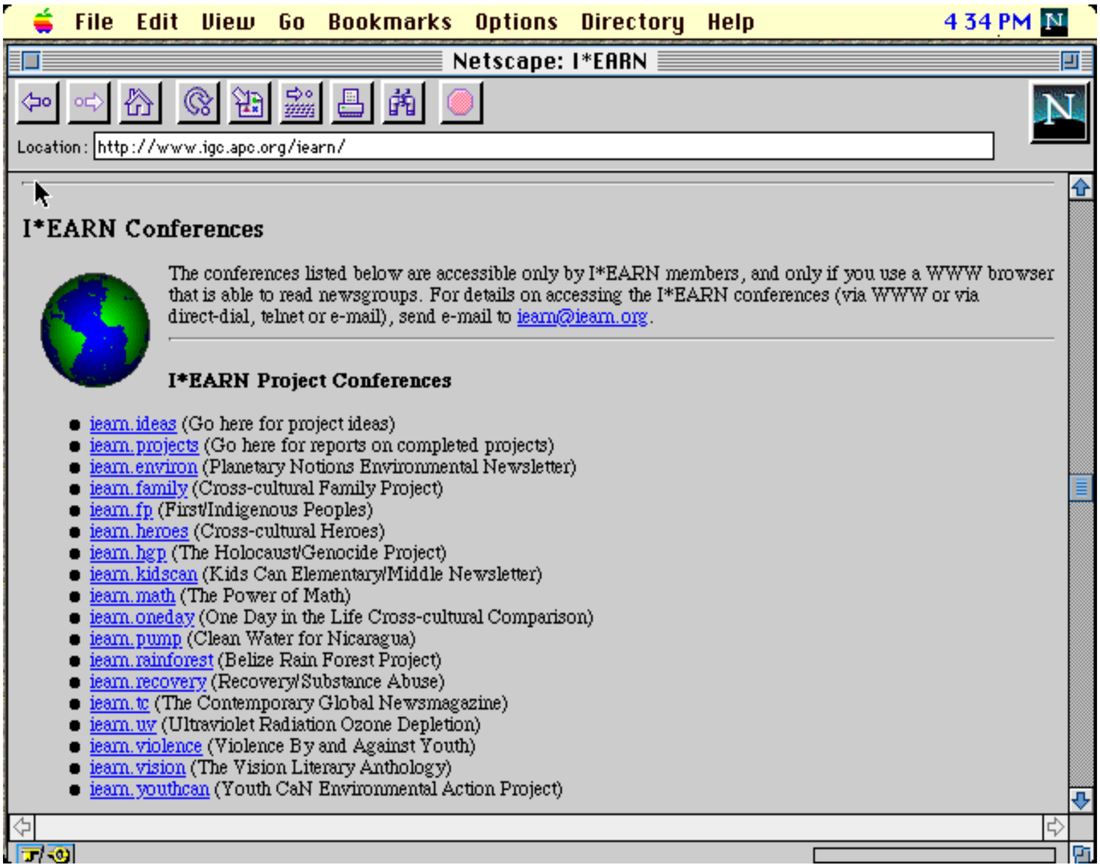

Figure 4: The I*EARN conference overview

The project-conferences are one of the most remarkable innovations of I*EARN. Until now the use of newsgroups has been mainly for debates or interchange of information. I*EARN now uses conferences to organize actions. This is a natural consequence of APC as a network for activists, that gathers 25.000 of them around the world.

When a teacher of the organization wants to launch a project, first he communicates the idea through e-mail to different people in the net, and after he receives some feedback, he will start a new conference to get more support to his initiative.

For the projects I*EARN also used a video-speaker telephone that gives a vivid realism to the connections. Finally they celebrate an international conference meeting once a year. In that reunion the world-wide I*EARN community meets all together.

As I*EARN gains support from their local authorities the local services change. For example, I*EARN Australia has a vivid Web server with different projects funded by educational authorities in the Victoria region.

The common language of the organization is mainly English. Spanish and other Latin languages are pushing for their acknowledgement too.

I*EARN allows students and teachers to work in different ways:

- in partnership with another class or youth group,
- with several other classes, who have expressed interest in a theme based project.

The theme based projects are the core activity of the network. At present they have launched a set of 33 projects in 4 main areas:

- environment/science/math (12),
- arts/literature (5),
- social studies/economics/politics (12),
- interdisciplinary areas (4).

This is how the I*EARN organization explains its own work:

> "Students and teachers work on collaborative projects primarily through threaded on-line conferences. Conferences are like rooms. Separate rooms are available for each of the projects. By going into the room, teachers and students meet each other to work on the same project together. Participants can enter the rooms at any time and read the discussion and see the project work that is underway"

There are three kinds of rooms. The first one is the "Meeting Room", called "iearn.ideas". In this room, teachers and students see the project ideas generated by others. By sharing ideas, the participants form new project teams and start new projects.

The second one is the list of projects done in the past. The third one is the I*EARN On-going Projects.

There are different kinds of projects:

- Scientific projects, like the First People Project, dedicated to the study of the culture of the First People around the World.
- Humanistic projects, like A Vision--An International Literary Anthology, a high school publication sponsoring student poetry, prose and artwork from around the globe.
- Technological projects, like Clean Water for Nicaragua Project, a co-operative effort with students to provide clean water for rural villages in Nicaragua.

Learning by projects seems to be a major characteristic of the I*EARN culture.

Recent studies (Fernando Hernández, Montserrat Ventura, 1992) have shown that the work projects are a new form of organising the school curriculum.

> "The main idea of the Projects...is that the students begin with the learning of procedures that allow them to organize the information, discovering the relations, that can be established from a topic or a problem"

Working by projects is changing a more traditional approach in education based on the so called centres of interest. A project is a new kind of organization of the educational activity. It gives the student a more active role in the design of his own learning process. Nevertheless until now these projects seemed confined to the school environment. They were projects inside the school.

The I*EARN projects seem to go a little bit further. They are based on the idea of building real solutions to real-life problems. In I*EARN words, "making a difference in the world". In this sense they are projects from the school to the society. I*EARN seems to think, that the students can help to solve actively this kind of problems through educational networks.

This philosophy could be resumed in two ideas:

a) Learning by doing projects and

b) doing projects not only for simulated problems, but for real-life problems.

For doing that, the computer network is not a random tool. Affordable communication of small secondary schools across the world is necessary for I*EARN projects. Secondary students are considered real global actors in the global information society.

These projects encourage I*EARN participants to form a "mutual support group". This group is in charge of designing the collaborative learning activities over the distance. Let us analyse some I*EARN projects.

The I*EARN Family Project(1994-1995)[11].

The Family Project has been one of the most successful activities in the net. As an initiative of the Spanish group of I*EARN this project got thousands of participants in 18 countries around the world.

---

[11] http://citel.upc.es:80/iearn/94f

Figure 5: The I*EARN Family Project

The project "La Familia en diferents països del mon" was an initiative of Narcis Vives, director of I*EARN Spain. The idea was to take advantage of the United Nations International Year of the Family to organize a school project in which each student and teacher could learn about the meaning of the family institution in different countries.

This local section of I*EARN has several distinctive features.

First, they stress the multicultural, multilinguistic projects. The Family Project intended to be a multilingual project in which every country could participate in its own language.

Second, the bottom up philosophy. The shared approach of the local section, leaded by a secondary teacher, is, that the projects must be a creation from the bottom up. Narcis Vives said:

> "Our idea is that the projects can grow up from the kids and teachers. That their own ideas can feed them."[12]

---

[12]Interview, May 11, 1995.

Finally, the cross-sectoral attitude. This section participates with other organizations in the creation of Pangea, a communication service for the international cooperation. In this service several professors and students of the UPC participate.

The Family Project was organized from an international meeting of I*EARN teachers celebrated in July 1994 in Puerto Madryn, Argentina. During several months this project was discussed through electronic mail between the main coordinators of I*EARN around the world.

Initially the project was designed around three topics-stages:

- My family and me
- Vacations and Holidays
- The family and the community old people.

These stages were divided in 5-7 weeks periods from September 1994 through April 1995.

Finally the project was organized around several proposed activities:

- The Affective Dictionary.
- The Change of Family Roles.
- Family Situations.
- Global Art Project.
- Survey on the marriage.
- Description and comparison of the families.

Maybe the most successful activity of the project was the building of the Affective Dictionary:

Each school was offered to send e-mails to the network to explain the meaning of a set of words related with the family (Hogar, Televisión, Padres, Abuelos, Separación, Apoyo, Muerte, Hermanos, Amigos, Familia, Castigo). The conference iearn.family collected 66 school messages from Chile (24), Catalunya(20), Argentina(13), Australia(7), Brazil(1) and Costa Rica(1). The messages were composed in Spanish (51), in Catalan (13), in English (8) and in Portuguese (1).

From an anthropological point of view this educational experiment has an enormous interest, especially for the ethnographic analysis of the family. This I*EARN project shows the meaning of "family" for children in several schools of different countries. It's true that the majority of answers described the nuclear western model of family, but there are remarkable exceptions. One school the Centro Provincial de Enseñanza Media Grupo "Lavandina" de Salta, Argentina explained their own conception of family in the Puna area with its own local terms. (e.g.Tatamama is their local word to describe family and it is composed by parents and grandparents). Other students from Bradford Secondary College in Australia explained their own version of family:

> "Families to me are a close-knit group of people, who stand by: each other, no matter what. Families don't necessarily have to be a mother, a father and kids. It could be two mothers and

kids or one father and a kid, the number of combinations is endless, but the basis is still the same–they're all families...".

It is remarkable how the television is considered a technology, that is part of the family universe. At the same time it is the only word, that collects contradictory feelings. For some it is considered an information and entertainment tool for the family ("Es donde miramos cosas que nos enseñan mucho", Escuela Campos Deportivos, Jovenes por la Paz, Temuco, Chile) and for the others is an annoying machine ("Es uno de los principales causantes de la discordia en la familia", Centro Provincial de Enseñanza Media. Argentina).

The second activity was the "Changes of Family Roles". Each student was invited to write a short story imagining he/she was another member of his family. 58 short stories were collected, all of them written from Catalunya, except one from Mansfield Secondary College in Victoria, Australia.

The third activity was "Situations in the Family" where the students described different daily situations in their respective families.

The result of the Family Project was more than 800 pages of e-mail and conference messages between the different schools and actors and the clear conclusion, that the project opened a new way of learning to the participants. From the technical perspective the tools used were very simple: e-mail, e-conferences and occasionally a video conference but at the global scale. Tools not too different from those used by the Internet Engineering Task Force to design new Internet protocols.

Problems in an Organizational Network.

Different problems now confront the evolution of the I*EARN, and they can be related with common problems in any global organizational network. Maybe the most important is the relation between global networks and national organizations.

From the educational point of view I*EARN is a highly innovative global network based on non-governmental organizations, but the national educational institutions are slow machines for educational reform. They are still constrained by traditional national curricula, traditional national practices and structures. These systems still organize the bulk of the young people at the different nations and they have the economic and administrative power. Changes in the educational curricula on the one hand need a more theoretical and long term debate between I*EARN and the academia. One of the main differences is the scope of educational curricula, global or national, based on non-governmental schools or based in the public educational system.

These differences could explain some local conflicts between I*EARN representatives and the administrators of national educational networks. In the USA for example, the National Science Foundation is sponsoring the Global Schoolhouse Project[13]. This is an initiative oriented to connect American k-12

[13] http://k12.cnidr.org/gsh/gshwelcome.html

schools to the Internet. It is a national effort for national schools very different to I*EARN ideas.

In countries like Argentina, the authorities have given to the I*EARN representatives the responsibility of organising the educational network nation-wide. The same in Australia, where the national organization is the responsible of the Global SchoolHouse Project[14]. At first sight, that could seem an appropriate way of introducing I*EARN in a country. But the tensions between national and global curricula, national and global structures, are not solved because of this alliance.

Finally, a last model is the case of Spain, particularly Catalonia, where I*EARN develops as a private, non governmental initiative, separated from the public school system. That explains the enormous difficulty to develop the network based only in voluntary efforts, but at the same time the originality and innovative scope of their proposals free of government tutorial.

Briefly, national organizations want to use the computer networks to modernize their practices and traditions. This is one way to analyse computer supported co-operative systems: conceiving them as new tools for old organizations. The other way is to begin analysing global computer and telecommunication networks like I*EARN, APC or Internet and the type of new organizations they are helping to create. Until now the research on CSCW systems has followed mainly the first approach. We think the second one can give a different perspective to the design of new computer supported co-operative work systems for new organizations, like I*EARN.

References

Bikson, T.K. and Gutek, B.A. (1983) Advanced office systems: An empirical look at utilization and satisfaction (N-1970-NSF).Santa Mónica, CA: The Rand Corporation.

Bowers, John, Ed. 'Introduction'. In Bowers, J.Ed. *A Conceptual Framework for Describing Organizations*. COMIC, D1.2 October, 1994.

Hammer, M. (1982) Improving business performance: the real objective of office automation, in *Proceedings of Office Automation Conference* (San Francisco):247-54.

Hernández, F. &Ventura, M. La organización del curriculum por proyectos de trabajo. ICE, Editorial Graó. Barcelona,1992.

Hiltz, S.R. and Turoff, M.(1978). The network nation: Human communication via computer. Reading, MA: Addison-Wesley.

Kiesler, S and Sproull, L. (1987). Computing and change on campus. Cambridge: Cambridge University Press.

Kling, R.(1980) Social analyses of computing: Theoretical perspectives in recent empirical research. *Computing Surveys*, 12, 61-110.

Kuutti,K & Virkkunen, J. 'Organizational Memory and the Learning Network Organization: Perspectives from Cultural Historical Activity Theory'. In Bowers, J. Ed. *A Conceptual Framework for Describing Organizations*. COMIC. D1.2. October 1994.

[14]http://www.peg.apc.org/~iearn/gclassroom.html

Serra, A. (1992) "Carnegie Mellon: An American Computer University". Session on Virtual Communities. 91th Congress of the American Anthropological Association. San Francisco, Dec 2-6.

Sirbu, M, Schoichet, S.,Kunin, J. and Hammer, M.(1981) *OAM: An Office Analysis Methodology*, Laboratory for Computer Science, MIT, Cambridge, MA.

Turner, J.A. (1984) Computer-mediated work: The interplay between technology and structured jobs. *Communications for the ACM*, 27(12),1210-1217.

Internet References.

Australian Global Classroom Project: http://www.peg.apc.org/~iearn/gclassroom.html.

Brown, Kristin (krbrown@igc.apc.org) (1995) Creating Global Learning Communities: I*EARN's Action-Based Projects. (http://aleph.ac.upc.es/HMP/PAPER/220/abst.html)

I*EARN Family Project(1994-1995). (Http://citel.upc.es:80/iearn/94f)

I*EARN Project: (http://www.igc.apc.org:80/iearn/)

U.S. National Science Foundation's Global School House Project: (http://k12.cnidr.org/gsh/gshwelcome.html)

Pangea Network: (http://pangea.upc.es).

Serra, Artur. Carnegie Mellon, An American Computer Campus. Session on Virtual Communities. American Anthropological Association. 91th Congress. San Francisco. Dec.2-9.1992. (http://www.ac.upc.es/~artur).

Chapter 5:
Work Processes, Organizational Structures and Cooperation Supports: Managing Complexity

Giorgio De Michelis
University of Milano, and RSO, Italy

Introduction

For many years service organizations (in both the private and public sectors) have tried to model their behaviour in accordance with the production process model, seeking improved efficiency, performance standardization, hierarchical control structures, et cetera.

Today the situation is radically different: some peculiar features of service processes have emerged, distinguishing them from production processes (Bowen and Schneider, 1988; Heskett et al., 1991; Fountain, 1993), meanwhile, products tend to embody value-added services (Takeuchi and Nonaka, 1986) and/or to transform themselves into services (Van Gorder, 1990). Also within organizations service relations are becoming the usual way to manage inter-departmental and/or inter-functional relations (Keen, 1991).

The emergence of service relations is due to the increasing request for personalized performances on the part of the customers; moreover, it reflects and/or sustains the growing complexity of organizational behaviours (Keen, 1991). Service relations can therefore be considered paradigmatic of todays organizations and of their evolution. The analysis of work processes in the context of service relations can offer a deep insight into the main features of some widely diffused organizational phenomena, along with some consistent and innovative guidelines for change management. Moreover, it can constitute a common, rich and innovative framework to all those studies aiming to define new grounds for the different disciplines studying the organization: from economy to social sciences, from management sciences to organizational theories, from psychology to computer supported cooperative work and information systems (De Michelis, 1995).

In this chapter the service paradigm is used to analyze and characterize work processes and their complexity. A work process, from this point of view, is characterized by the communicative relations binding its participants and embedding their performances.

The complexity of a work process is then related both to its value (the external complexity) and to its cost (the internal complexity). It is shown that in order to face great external complexity with small external complexity it is necessary to enhance the knowledge creation process.

Knowledge creation processes are then related to organizational design issues, to the empowerment of professional skills and to the computer support systems capable of helping people to manage the complexity of their work effectively.

Work Processes, Communities of Practices and Social Complexity

A service is generally characterized in the literature as being something that is not "a good" (Fountain, 1993). Between pure services and pure goods lie all those processes where goods are enriched by something that is not a good and where non-goods embed some goods: a clear dichotomy between them is therefore an oversimplification, and service can be considered as a viewpoint for observing any work process.

Despite the variety of categories into which they can be classified, all services have three fundamental characteristics: intangibility, customer co-production, and production/consumption simultaneity (Bowen and Schneider, 1988). Service intangibility conveys the idea that a service relation results in the experience having taken place rather than the object having been transferred. The value of a service is highly subjective, since it depends on its customers. Service co-production conveys the idea that the customer is not passive in the service relation. The unique temporal proximity of production, delivery and consumption in a service relation implies that quality control must take place in real time and that front office agents are crucial to assure the quality of the delivered services.

The three above characteristics can be considered as the main aspects of the service paradigm and they can be used to observe the complexity of work processes. as social processes.

From the service point of view work appears as a process relating its performer (the singular form is an abstraction that can be accepted at this point: generally a work process has more than one single performer; this point will be discussed in the following pages) to the customer (idem) who recognizes its value. While performing a work process the performer consumes resources and generates a value for the customer. The customer is the legitimate beneficiary of the performance of the work process (no distinction between production, delivery and consumption is meaningful from this point of view; see service characterization above) and the value she attributes to it has a social recognition.

The value of a work process is therefore intangible and subjective (see service characterization above). The social recognition of the value of a work process can, in fact, be given by means of the price the customer pays for it, but it can also be given by the resources she consumes (within a work process the performer and the customer perform together, they co-produce: see service characterization above), or finally by her declaration of satisfaction. All the three above behavioural patterns exhibit in some sense the appreciation of the customer for the performed work: it is up to the performer to determine if the socially recognized value of her work is sufficient with respect to her cost (with respect to the resources she has consumed).

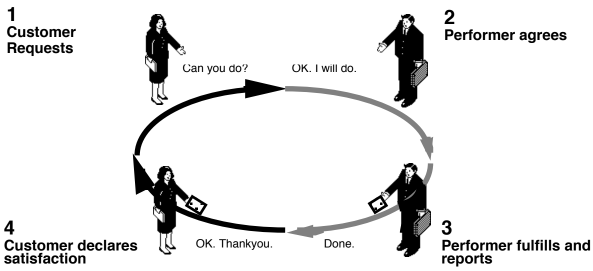

Figure 1. The Action Workflow Model - ATI™.

No matter the means through which the value of a work process is recognized, it is clear that the value recognition is carried on through the communicative interactions between the performer and the customer occurring during the work process. Through communicative interactions the performer and the customer reach, in fact, an agreement on the actions to be performed and share the evaluation of their execution. The actions performed within a work process are embedded into the communicative interactions between its customer and its performer.

To work is therefore not simply to act with a purpose: what a person does can be considered work if she performs something in order to satisfy a request made by another person[15]. Work processes - see Figure 1 - embed actions within a relationship, whose essential nature is communicative and pragmatical[16].

The relationship between a performer and her customer is basic within a work process, but customer - performer relationships are not the only ones relevant within it. A work process involving only one customer and one performer is in fact a simple limit case (many observers call it individual work - for example (Schmidt

[15]The above definition of work is due to a personal rethinking (De Michelis, 1994, 1995) of the language/action perspective as it has been proposed by Fernando Flores, Terry Winograd and co-workers (Flores, 1982; Winograd and Flores, 1986; Medina Mora *et al.,* 1992).

[16]The customer - performer relationship has been inspired by the Action-Workflow™ (Medina Mora *et al.,* 1992).

and Bannon, 1992) - as there is only one performer). Generally work processes are more complex: they involve several customers and several performers, belonging to different organizations (Figure 2), and they can be decomposed into sub-processes within which some actions are performed necessary to the completion of the main work process (Figure 3).

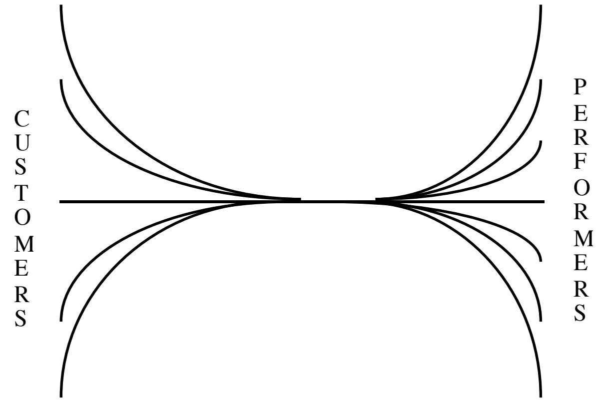

Figure 2. A work process as a relation involving several customers and several performers.

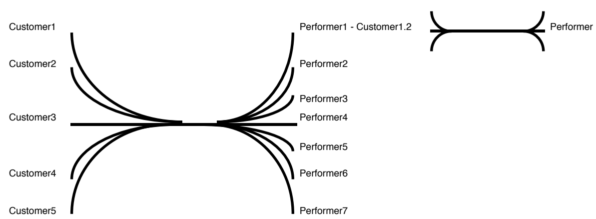

Figure 3. A work process with one of its sub-processes.

The relationships between the actors of a work process can be characterized from the point of view of the positions they occupy within it. Beside the basic relationship between the customers and the performers of a work process, the relationships between its customers as well as the relationships between its performers have also to be taken into account, and the relationships between actors of different sub-processes of the same work process, since any positional relationship type within a work process is characterized by a specific form of cooperation between its participants. Let us present them briefly here below[17].

Collaboration is the cooperation form occurring between the customers and the performers of a work process. It is the basic form of cooperation within work processes, since value is created through it. Collaborating persons interact doing something together. Successful collaboration creates a common language and a

[17]For a detailed discussion of cooperation forms see: (De Michelis, 1994).

common understanding between the customers and the performers through their mutual listening.

Codecision with equal roles is the cooperation form occurring between the customers of a work process. It logically precedes the action: if it is carried out successfully, it fixes the common expectations with respect to it. Codecision with equal roles is a decision process where all the participants share full responsibility for the decision to be taken.

Coordination is the cooperation form occurring between persons being actors within different sub-processes of a work process. It allows the integration of the different sub-tasks that are parts of a more general common task. Successful coordination depends on the degree of awareness of their organizational context of its participants.

Codecision with distinct roles is the cooperation form occurring between the performers of a work process. Through successful codecision with distinct roles the participants in a work process recognize and maintain the mutual interfaces between their respective performances. It is a decision process, where each one of the participants takes a decision on the issue on which she has responsibility within the constraints defined by the decisions of the other participants.

The participants in a work process share an experience constituting them into a whole, into a community. This community has all the features that characterize the communities of practices, studied by work anthropologists[18], since its members share a space (physical and/or virtual; this point will be deepened later), some artifacts (tools and resources), a language, the knowledge of the world they live in and of the possibilities it opens, the history of the work process to which they participate and the value they create within it.

As a social phenomenon a work process is complex: the community of practices involved in it is a network of social relationships that cannot be reduced to any functional model. Despite any attempt to plan its evolution with respect to its expected outcome, its history is unpredictable. The members of a community of practices during a work process in fact change their understandings, their requests, their ways of performing, their mutual agreements in a history of successes and failures, in a common experience of learning and knowledge creation, in the building of their community bonds.

Managing a work process in a normative way constraining its participants to some predefined plans is, therefore, ineffective: as the service literature points out, effective management cannot be disjoint from change management (Heskett *et al.,* 1990). Within a work process the unexpected events, the breakdowns, play a central role: on the one hand, increasing costs, menacing the possibility of satisfying customers; on the other, activating the learning process, updating ineffective plans, exploiting new opportunities.

Within a community of practices various breakdowns may occur, either generated by the changing conditions of satisfaction of the customers, or by the

[18]See the contribution of Susan Stucky to: (Bagnara *et al.,* 1995).

difficulties the performers meet in order to satisfy them. In both cases, breakdowns can generate new breakdowns, either when their solution goes beyond the capabilities of the members of the community of practices, or when they fail to cooperate in accordance with the positional relationships they occupy in the work process. The breakdowns of a work process depend on its complexity and reflect it: if a work process is simple it does not create surprises.

Complexity is a rather fuzzy concept, characterizing in some cases the difficulties an observer encounters facing a phenomenon, in other cases, a quality of the phenomenon itself. The second case implies the first one, while the converse is not true. In this chapter complexity is therefore characterized as the quality making a phenomenon not reducible to any explicative model. The complexity of a work process can be considered, from this point of view, as the combination of two factors: its multiplicity and its autonomy.

The multiplicity of a work process characterizes its complication: it depends on the number of its actors (customers, performers) and of the relationships binding them, on its duration, et cetera. The autonomy of a work process characterizes its unpredictability: it depends on the quality of the professional skills of its actors, on the difficulty of their tasks, on the degree of parallelism of their perfomances, on the frequency of breakdowns occurring to them, et cetera.

Pure multiplicity can be controlled through well structured plans; pure autonomy can be controlled through close observation of everyday behaviour (single persons autonomy can be approximatively anticipated); but the combination of the two is not controllable, since autonomy imposes frequent plan changes, and multiplicity hinders close observation (any person sets her own pace, has her aims, her image of the world, the images of any relationship are multiplied by its observers, by the different timings of its observation, et cetera).

The community of practices performing a complex work process is therefore constituted by several actors with high competence, facing non routinary problems and interacting in a rather creative way. The complexity of a work process affects both its value (with low complexity only small values can be created) and its cost (high complexity consumes time and other types of resources). Managing a complex work process requires containing its cost without affecting its value. The cost of a complex work process is not predictable through the traditional accounting methods of traditional management. It is constituted more by transaction costs (Williamson and Winter, 1993), i.e. by the cost of the interactions between the actors, than by production costs (the cost of the actions performed by the actors).

If we distinguish external complexity, generated by the customers and their requests (qualifying the value of the work process), from internal complexity, generated by the performers and the requests they do to other persons in order to perform their tasks (determining the amount of time and other resources necessary to perform it), the quality of a work process can be characterized by its capacity to face great external complexity with small internal complexity.

It seems a paradoxical objective, since it is a widely diffused opinion that there is a direct correlation between external and internal complexity[19]. It is well known, on the contrary, that different communities of practices exhibit different effectiveness facing a problem: where one community exhibits all the needed competences ready at hand, a smooth way of interacting, a prompt mutual understanding and performances fitting the requests, another one is continuously fighting against misunderstandings, missing persons, lack of competences, wrongly focused performances.

Despite the direct correlation between the internal and external complexity of a work process, different communities of practices are characterized by a different correlation parameter. The quality of a work process depends on it.

What is the factor determining a low value of this correlation parameter? The factor allowing to face a growing external complexity without letting the internal complexity grow consequently? Taking into account the dynamic nature and the impredictability of a complex work process, this factor has to do with the transparency and/or visibility of the community of practices performing it, with the richness and flexibility of its communication network, with the quality of the competences of its members and, finally, with its plasticity with respect to the external complexity of the work process.

The analysis of work processes proposed so far indicates a good candidate for playing this role: it is the knowledge created and shared by the community of practices performing a work process through its individual and collective learning practice (about the work process history and expected outcomes, about the competences needed to perform it, about the community of practices performing it, its language, the space it occupies, the artifacts available in it, its members, their competences, their mutual relations, et cetera).

Knowledge creation processes within organizations have been characterized by I. Nonaka (1988, 1991, 1993) in terms of tacit and explicit knowledge transformation: he considers the knowledge transformation processes the means through which, in its evolution, the individual knowledge of any member becomes knowledge shared by the whole community and, conversely, the knowledge of the whole community projects itself into the individual knowledge of the members. Knowledge transformation processes are Organizational Learning processes performed by autonomous and responsible persons: they can be associated to the above presented cooperation forms, since any type of knowledge transformation is performed through a specific type of interaction, i.e. through a specific form of cooperation, between the members of a community of practices (De Michelis, 1995b).

Good cooperation is the principal means through which communities of practices create knowledge, through which they learn. Managing complex work processes is a radically different matter with respect to traditional accounting management: its problem is to enhance and direct the learning capability as well as the learning

---

[19]Cybernetics has stated the requisite variety law on this issue. See for example: (Ashby, 1956).

practice of the community of practices performing a work process towards customer satisfaction[20].

Knowledge creation processes are strengthened by any change reducing the obstacles to effective cooperation (e.g.: simplifying the process plan, making control easy, designing process-oriented organizational structures) and enhancing the learning capability of the community of practices (for example: creating responsible roles, designing adequate work spaces, providing adequate tools).

The simplification of the work plan as well as the design of easy control systems are actions that are consistent with the Business Process Reengineering (BPR) approach (Hammer, 1990; Hammer and Champy, 1993; Davenport, 1993), since they cancel all the non necessary intricacies making the flow of actions difficult and therefore creating a confused background to the learning process.

Professional empowerment (see Section 3) as well as organizational design (see Section 4) issues integrate the BPR approach with the Organizational Learning one proposing a revisited Socio-technical System Design approach (Butera, 1994), taking into account not only the need for improved process performances but also the need for preserving the value generated by previous experiences.

The failures of many BPR projects (for example, Gartner Group claims that 50 to 70% of BPR projects fail to meet their original objectives) have been attributed to their lack of paying attention to human and organizational problems (Davenport, 1994; Hall *et al.*, 1993): a revisited Socio-technical System Design approach allows to overcome those limits through a more comprehensive and effective change management.

Within this context, also the role of computer support systems for work processes has to be considered again, since they can play a relevant role in reducing the internal complexity of work processes (see below, in particular Section 5).

Professional Skills within Communities of Practices

The members of a community of practices are persons sharing a meaningful common experience: their effectivenes within the work process they perform together depends on their capability of playing a responsible role in it. A responsible role requires not only mastering the domain knowledge characterizing the tasks assigned to it, but also the capability of interacting with the customers and with the other members of the community of practices, creating value with them. The members of a community of practices, therefore, are not only specialized technicians but also organization professionals (Butera, 1994) capable of performing for customer satisfaction. The growing complexity of the work

[20] This observation rephrases Organizational Learning imperatives; see: (Argyris and Schoen, 1978; Senge, 1991).

processes to which they participate requires, among other, that they are able to manage the breakdowns which may occur without creating further complexity.

Any person, as anyone has probably experienced, is capable of managing a certain degree of complexity, beyond which her perfomances lose effectiveness: she is overwhelmed by her duties, she is continuously reordering her agenda, modifying the commitments she has with other people, she fails to do what the other expect from her. Let us call this behavioural limit of any person her threshold of sustainable complexity (De Michelis, 1994).

A person that has surpassed her threshold of sustainable complexity is provoking a growth of the internal complexity of any process to which she participates, since she is continuously generating new breakdowns, involving the other actors of the work process with whom she cooperates without contributing to solve any of them. The threshold of sustainable complexity of any person is not fixed once for all, it evolves, on the basis of her experiences and it depends on various interconnected factors.

As first, it is affected by the eventual asymmetries between the tasks she has to accomplish and her competences: if a person has not the professional competence to perform her tasks, if she is not capable to act in a responsible manner, then her threshold of sustainable complexity decreases, since the difficulties she encounters to accomplish her duties are stressing her behaviour so that she loses effectiveness and is paralyzed by the eventual breakdowns.

As second, it depends on her awareness of the work process she is participating and of the community of practices performing it: if she does not know what has happened and what is going on, if she does not know the positional relationships binding her with the other members of the community, if she does not know their professional competences, if she does not know where are the resources, the tools, the information, the persons she may need, then her threshold of sustainable complexity decreases, since she has not ready at hand the context within which she is acting and inter-acting.

As third, it depends on the nature of the space she shares with the community of practices whose she is member of and on its tool equipment. The locations where the members of a community of practices work (no matter if they are physically adjacent or distant) are integrated by the media they can use to communicate each other: the electronic media interconnect the physical locations creating the unique virtual space of the community of practices. If the space where a community of practices works, offers a rich set of communication possibilities to its inhabitants, if it is capable to bring forth the context of any interaction they may perform each other, if it provides efficient and highly usable tools ready at hand when necessary, then it increases their threshold of sustainable complexity, since it reduces the effort they need for managing the complexity of the work processes they are performing. Computer support systems are today the main components of the space of a community of practices and of its tool equipment, but they are not alone: their design must exploit their individual functionalities as well as their integration in the physical space.

The three above factors are mutually inter-dependent, since they are mutually enhancing each other: the professional skill and the awareness of the work process to which a person is participating are in fact increased by the quality of the space where she is acting and of its tool equipment, while poor professional skill and poor awareness do not allow a full exploitation of the services offered by the space and its tools.

The threshold of sustainable complexity characterizes the individual factors affecting the learning capability of a community of practices: increasing it means empowering its members from the point of view of their capability to create organizational knowledge.

Organizational Structures for Process-Oriented Organizations

The work process point of view presented in Section 2 offers an original account on how organizations emerge within complex work processes.

The knowledge created within a work process can be analyzed in two different contexts. On the one hand, it allows to reduce the internal complexity the community of practices performing it will create in any new work process in which it engages itself; on the other, it allows to reduce the internal complexity its performers (the persons participating to it in the performer position) will create in any new work process of the same type in which they engage with other customers. While the creation of knowledge shared by the whole community of practices characterizes the growing partnership relations binding each other its members, the creation of knowledge shared by its performers (or by some of them) characterizes their increasing efficiency and effectiveness in performing a particular type of work process, i.e. their increasing capability to perform consuming less resources and creating greater value.

In order to reduce the resources (the cost) of their performances, the performers enforce their being performers, separating themselves from the customers and trasforming their experience into (practical) knowledge usable within other work processes (with other customers): they create an organization on the performer side.

Organizations emerge as means for trasforming the experience some persons do performing a work process into a structure enabling them to increase their efficiency and/or effectiveness. An organization gives a collective identity to its members: defining membership conditions; characterizing their roles (both in terms of responsibility and competence) and the products/services they deliver; providing them with work spaces, with material, technical and/or financial resources, with behavioural patterns (procedure templates, forms, ...), with professional information and, last but not least, with a language through which they can share all those things and give sense to what they do.

The necessity of restoring efficiency and effectiveness is frequently reappearing in organization today, since the turbulence of the environment continuously improves the external complexity of work processes. While working within work processes the organization continuously evolves in order to try to restore effectiveness. The process of working and the process of maintaining the organization are (mutually related) autonomous processes whose coupling can be granted only in dynamic terms: the latter in fact involves an interpretation of what is going on within the former as well as of the organization itself and implies the creation of a shared understanding. Since language allows the members of an organization to give sense to their behaviour within various organizational metaphors (Morgan, 1976), sharing is more a matter of practice than of semantics: the organization is performed (Bowers, 1993) by its members.

It cannot be taken for granted that an organization is performed in such a way that its effectiveness is restored: the history is and will be full of organizations which fail to do this: maintaining an organization, in fact, is not a deterministic process; its results can be different from its aims. The changes an organization introduces while its members perform it in fact mediate between its current structure and its new needs, between its current language and knowledge and the improved performances required by the environment: performing an organization has the aim of reducing its costs but it is exposed to the risk of increasing its internal complexity, and consequently to fail its objective.

The work process point of view we have sketched in the second Section suggests, in order to avoid the risk of increasing through organizational change the disalignment between the value and the cost of the performances of an organization, to increase its learning capabilities, to improve its knowledge creation processes.

The organizational units (offices, departments, et cetera) are, from this point of view, the repositories of the knowledge created by an organization: its units of an organization give duration and stability to the knowledge creation process and, moreover, make that knowledge available for other members and other units of the same orgnaization. Where pure adhocracy has been implemented, creating and sharing knowledge becomes difficult and unsafe. Organizational units are not the teams performing a work process: while the latter ones have the objective to improve their performances to meet customer needs, the former have the objective to maintain and develop the knowledge that is created and used within work processes.

Within an organization performing complex work processes we can distinguish three different types of managerial roles: (1) the work process owner, who is responsible of the performances of a class of work processes: her focus is on the customer needs, she is, in some sense, an internal customer of the professional (and material) resources of the organization; (2) the responsible of a group of professional (and/or material) resources: she is responsible of maintaining and increasing their value, i.e. of exploiting their learning capabilities, of enhancing their knowledge creation processes; (3) the executive who designs the business of the organization, balancing the quality of its performances, the value it offers to its customers, with the exploitation of its resources, with its internal value.

Since the success of an organization depends on the cooperation between the three above sketched managerial roles, on their ability to maintain their own knowledge creation process, it is necessary that their mutual relationships are defined in terms of mutual services, and not in terms of pure hierarchy. Learning organizations are always more similar to network organizations (Butera 1993). Computer support systems are very important to let a network organization perform effectively, since, creating and equipping the virtual space where its managers cooperate (the electronic network is in some sense a model of the organizational one), it helps them to sustain its internal complexity.

Computer Support Sysytems for Process-Oriented Organizations

Information and Communication Technology (ICT) has been widely applied from the mid of the sixties as a mean for reducing the costs of organizations: as the Socio-technical school has pointed out from those early years, its introduction within an organization has to be considered as part of its process of organizational change. The Information Systems that were developed up to the late eighties were mainly oriented to increasing the efficiency of the organizations and not to reduce their internal complexity. It is a popular complaint today that Information Systems failed. The above observation contributes to explain why: efficiency does not suffice to meet customer needs and to improve performances.

In the late eighties a new family of organizational computer-based system and a new way to conceive the use of ICT within organizations appeared, orienting their application in a rather different direction with respect to traditional Information Systems: ICT is proposed as a means of supporting the cooperation of the persons involved within work processes and not as a means of substituting them. Computer Supported Cooperative Work (CSCW) is the multidisciplinary research area facing the problem of analyzing, designing and evaluating cooperative work processes as well as their support systems (AA.VV., 1986, 1988, 1990, 1991, 1992, 1993, 1994; Bowers and Benford, 1991).

CSCW systems (called also Groupware (Ellis *et al.*, 1991) or Workgroup Computing Systems) are the systems aiming to support people cooperating within work processes and therefore our attention can now focus on them. Among the systems and prototypes developed within the CSCW field there are Workflow Management Systems, Knowledge Sharing Environments, Conversation Handlers, Group Decision Support Systems and various multimedia communication supports (videoteleconferencing systems, computer supported meeting rooms, ...). The variety of the proposed systems is very rich, but CSCW systems are far from having reached a steady state, where offered services, underlying architectures and human interfaces can be characterized in accordance with some standard categories. Moreover, the relationships between CSCW systems and the traditional Information Systems have still to be investigated and fully understood. It is very difficult therefore to define ways to choose and/or design the support system needed by an organization on the basis of the work processes it performs and the qualities it needs while performing them.

The conceptual framework developed so far offers some hints in this direction, that are briefly sketched below. From the work process point of view, computer support systems are, on the one hand, artifacts equipping the space where a community of practices works, on the other, means through which that space is enhanced to a plastic dynamic virtual space. Moreover, computer support systems can contribute to reduce the internal complexity of the work processes incresing the threshold of sustainable complexity of its actors and creating the electronic network giving shape to the organization performing it.

a) Workgroup computing systems are not tools to process information, but enabling artifacts. Information processing is the way through which computer based systems enable their users, not the service they offer to them. Through information processing, in fact, they can do many things that are relevant to help them to sustain the complexity of the work processes where they are acting: they can make available the knowledge created and/or shared within a work process; they can automatically maintain the multimedia information basis of the process; they can offer a knowledge basis modelling the organizational environment. In all these sample cases their utility depends on the transparency and/or visibility they offer to their users about the organizational structures, the work processes, the communities of practices in whose context they perform. It is therefore very important that they are ready at hand when necessary: designers should pay attention to the way they are physically designed and located.

b) Workgroup computing systems transform the physical space into a virtual space supporting multimedia communication. Their quality depends on the multiplicity of communication media they support: if they support both synchronous and asynchronous communication; if they link together sequences of interrelated communication events; if they link effectively communication to action, then the virtual space they create allows its inhabitants to cooperate effectively independently of the time and space conditions constraining them.

c) Workgroup computing systems absorb routinary tasks and dissolve pure hierarchical control supporting the workflow. If they create automatically distributed to-do lists, if they integrate any type of productivity tools, if they keep track of the work done, if they are able to support exception handling, then they create a work setting where the actors can exploit their capabilities and responsibilities.

d) Workgroup computing systems support the awareness of their users, as much as they make visible and/or transparent the knowledge created within work processes, as they reflect their changes. Thank to the success of Internet workgroup computing systems can create an open virtual space where changes of membership, of roles, of relationships are adequately reflected.

At the Cooperation Technologies Laboratory of the Department of Infomation Sciences of the University of Milano we are currently developing a prototype of a system supporting cooperation within work processes, that is inspired by the above remarks (De Michelis and Grasso, 1994; Agostini *et al.*, 1995). Its aim is to show the feasibility of a system with the above characteristics.

Conclusion

This chapter proposes a conceptual framework based on the analysis of work processes and their complexity to analyze and design organizational behaviour. It offers some first hints on how to characterize organizational behaviour from the point of view of the distinction between internal and external complexity and of knowledge creation processes (Nonaka, 1993). It sketches some observations about the professional skills, the organizational structures and the computer support systems well suited for process oriented organizations. But any issue raised by it requires further investigation and discussion.

Its main claim is that the analysis of work processes and of their complexity, on the one hand, can be the basis for updating the Socio-technical System Design approach with respect to new popular change management methods as Business Process Reengineering, Organizational Learning, Professional Empowerment and Workgroup Computing Systems, avoiding the limits of each one of them, on the other, can offer a renewed unified ground for a paradigm shift of the various disciplines studying the organizations.

With respect to both these issues this chapter is only posing the problem and indicating a research direction.

Acknowledgments

This chapter presents a research that has been conducted with the financial support of the EC within the ESPRIT- BRA Project 6225, COMIC and within the IMPACT project of the COST-14 Action. Alessandra Agostini and Maria Antonietta Grasso of the Cooperation Technologies Laboratory of the Department of Information Sciences of the University of Milano deserve a particular credit for their contribution to the development of the ideas proposed in the chapter and for their careful reading of a preliminary release of it. The professional collaboration of the author with his colleagues at RSO, mainly Federico Butera, Thomas Schael and Buni Zeller, has also influenced him deeply.

References

AA.VV. (1986). Proceedings of the Computer Supported Cooperative Work Conference 1986. MCC, Austin.

AA.VV. (1988). Proceedings of the 2nd Computer Supported Cooperative Work Conference 1988. ACM, New York.

AA.VV. (1990). Proceedings of the 3rd Computer Supported Cooperative Work Conference 1990. ACM, New York.

AA.VV. (1991). Proceedings of ECSCW'91. Kluver, Dordrecht.

AA.VV. (1992). Proceedings of the 4th Computer Supported Cooperative Work Conference 1992. ACM, New York.

AA.VV. (1993). Proceedings of ECSCW'93. Kluver, Dordrecht.

AA.VV. (1994). Proceedings of the 5th Computer Supported Cooperative Work Conference 1994. ACM, New York.

Agostini A., G. De Michelis, M. A. Grasso, W. Prinz and A. Syri (1995). Contexts, Work Processes, Work spaces, Computer Supported Cooperative Work. An International Journal, (to appear).

Argyris C. and D. A. Schoen (1978). Organizational Learning. Addison-Wesley, Reading.

Ashby R. (1956) An Introduction to Cybernetics. Chapman & Hall, London.

Bagnara S., S. Stucky and C. Zucchermaglio, Eds. (1995). Organizational Learning and Technological Change. Springer Verlag, Berlin.

Bowen D. E. and B. Schneider (1988). Service Marketing and Management. In: Research in Organizational Behaviour, Vol. 10, JAI Press.

Bowers J. (1993). Understanding Organization Performatively. In: Issues of Supporting Organizational Context in CSCW Systems (L. Bannon, and K. Schmidt, Eds.), COMIC D1.1, pp.49-72 (Available via anonimous ftp from: ftp.comp.lancs.ac.uk).

Bowers J. and S. Benford, Eds. (1991), Studies in Computer Supported Cooperative Work. North Holland. Amsterdam.

Butera F. (1994). Network of Enterprises or Network Enterprise? In: Proc. of CEFRIO Conference, University of Quebec Montreal.

Davenport T. H. (1993). Process Innovation: Reengineering Work Through Information Technology. Harvard Business School Press, Boston.

Davenport T. H. (1994). Saving IT's Soul: Human-Centered Information Management. Harvard Business Review, **72**.2.

De Michelis G. (1994). From the analysis of cooperation within work-processes to the design of CSCW Systems. In: Proceedings of the 15th Interdisciplinary Workshop on Informatics and Psychology: Interdisciplinary approaches to system analysis and design, Schaerding, May 24 - 26.

De Michelis G. (1995). Computer Support for Cooperative Work: Computers between Users and Social Complexity. In: (Bagnara et al., 1995).

De Michelis G. (1995b). Work, Knowledge and Computer Support Systems. unpublished notes.

De Michelis G and M. A. Grasso (1994). Situating conversations within the language/action perspective: the Milan Conversation Model. In:
(AA VV 1994), 89-100.

Ellis C. E., S. J. Gibbs and G. L. Rein (1991). Groupware: some issues and experiences. Comm. ACM, **34**.1, 39-58.

Flores F. (1982). Management and Communication in the Office of the Future. Hermenet, San Francisco.

Fountain J. E. (1993). A Customer Service Literature Review. In: Customer Service Excellence, JFK School of Government - Harvard University, Cambridge.

Hall G., J. Rosenthal and J. Wade (1993). How to Make Reengineering ReallyWork. Harvard Business Review **71**.6.

Hammer M. (1990). Reengineering Work: Don't Automate, Obliterate. Harvard Business Review, **68**.4, 104-111.

Hammer M. and J. Champy (1993). Reengineering the Corporation. Harper Business, New York.

Heskett J. L., W. E. Sasser Jr. and C. W. L. Hart (1990). Service Breakthrough: Changing the Rules of the Game. Free Press, New York.

Keen P. G. W. (1991). Shaping the Future: Business Design through Information Technology. Harvard Business School Press, Boston.

Medina-Mora R., T. Winograd, F. Flores and R. Flores (1992). The Action Workflow Approach toWorkflow Management Technology In: (AA VV, 1992), 281-288.

Morgan G. (1986). Images of Organizations. Sage Publ., Newbury Park.

Nonaka I. (1988). Toward Middle-Up-Down Management: Accelerating Information Creation. Sloan Management Review, **29**.3.

Nonaka I. (1991). Managing the Firms as Information Creation Process. In: Advances in Information Processing in Organizations (J. Maindl, Ed.), Vol. 4, JAI Press.

Nonaka I. (1993). On a knowledge creating organization. Paper presented at the XII AIF Conference, Parma, Italy, October.

Schmidt K. and L. Bannon (1992). Taking CSCW Seriously: Supporting Articulation Work. Computer Supported Cooperative Work. An International Journal, **1**.1/2, 7-40.

Senge P.(1991). The Fifth Discipline: the art and practice of the learning organization. Doubleday Currency, New York.

Suchman L. A. (1987). Plans and Situated Actions. Cambridge University Press, New York.

Takeuchi H. and I. Nonaka (1986). The New New Product Development Game. Harvard Business Review, **64**.1.

Van Gorder B. E. (1990). Total Customer Service - The Real Competitive Edge. Credit, **16**.6.

Williamson O. E. and S. G. Winter, Eds. (1993). The nature of the firm. Oxford University Press, New York.

Winograd T. and F. Flores (1986). Understanding Computers and Cognition. Ablex, Norwood.

Chapter 6: An Object-Oriented Framework for Organizational Modelling

Wolfgang Prinz
GMD, Germany

Introduction

The modelling of complex information systems requires the consideration of different modelling views which reflect different aspects of the information system. This section introduces a framework that provides different modelling views of the TOSCA organizational information system. The components of this framework will be discussed in detail by the following sections of this chapter.

The TOSCA modelling views

The following figure presents the different model views of the organization information system.

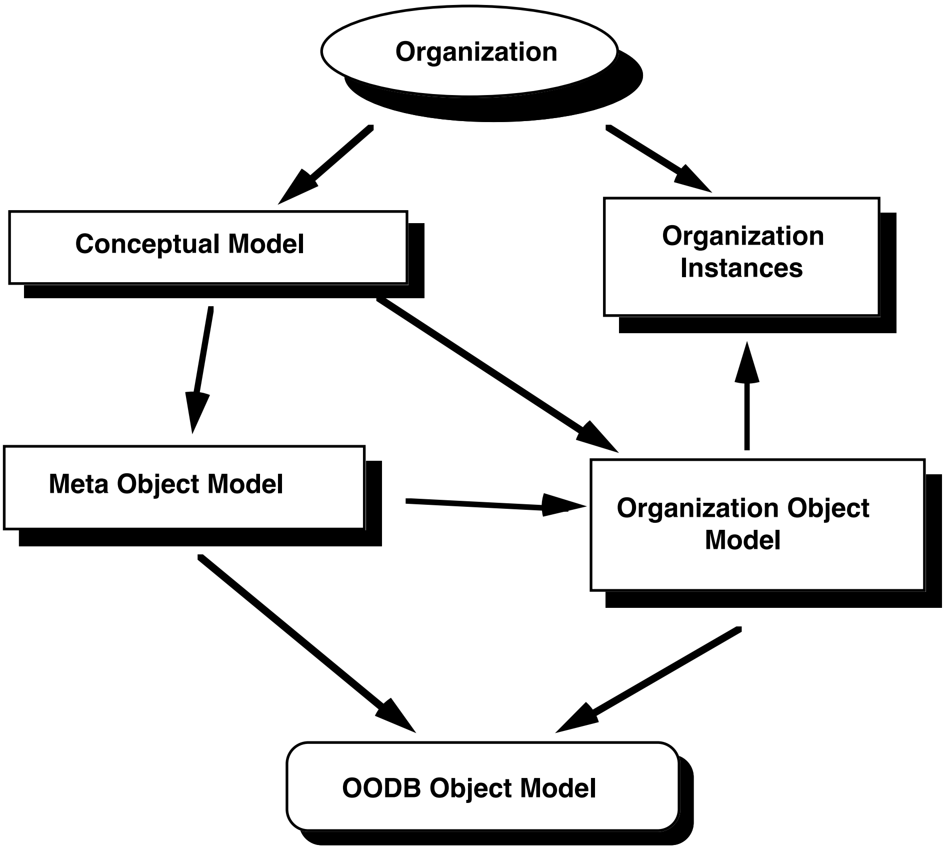

Figure 1. The TOSCA framework for organizational modelling showing the different components and their dependencies

The top level of the figure represents the 'real world' **organization**. That is the organization that is to be modelled and represented by the organization information service (OIS). The general term organization is understood to include different kinds of work organization. On the macro level it may describe large organizations such as enterprises or the organization, of a project or committee on a micro level. It is also possible that the structures and components of so-called virtual organizations are represented, i.e. of organizations which are created and maintained solely by the provision of electronic communication media, or virtual reality systems.

The **conceptual model** component identifies and describes the concepts in the organization model, i.e. organizational entities (e.g. department, project, role) and their relationships (e.g. reports to, belongs to, responsible for) on an abstract level. The related work review has shown that semantic nets are a common and useful modelling technique for the description of the conceptual model of a particular organization.

The **meta-object model** defines the object model for the organizational representation. It defines the object representation of the two basic components of the semantic net, i.e. the organizational entities and relationships. Furthermore, this level includes the definition of an event object model which is used as a mechanism for the provision of awareness of organizational changes.

The specific organizational entities and relationships identified in the conceptual model are defined as object classes in the **organization object model**. Here the properties and behaviour of the identified concepts are defined. The definition of the appropriate object classes follows the specifications of the meta-object model. The object classes defined here depend on the specific organization that is modelled, while the meta-object model is organization independent. Nevertheless, it is intended that the basic classes of the organization object model represent a toolbox that contains classes which can be applied to a variety of organizations with minor subsequent specialisations.

For the storage and maintenance of the organizational information, the meta object model must be mapped onto the data model of the underlying database. With the decision to choose an object-oriented model for the OIS, this is ideally an **object oriented database**. But even in this case a mapping will be required, because the general-purpose database object model will not correspond to the application specific OIS meta-object model in a one-to-one fashion.

The organization object model forms the basis for the creation of **organization object instances**, which actually represent a specific organization with its components and structures.

The following sections describe the organization, conceptual model, and meta-model components of the TOSCA framework in more detail. The organization object model is presented in chapter 6 of this deliverable, the application of the OIS by a CSCW application is presented in chapter 7.

Considered Organizations

It is the objective of the OIS presented here to be applicable to different kinds of organizations. Thus the organization component in figure 1 represents different organizational forms and different insights into organizations. In this context we can distinguish between a horizontal and vertical dimension which are illustrated in figure 2.

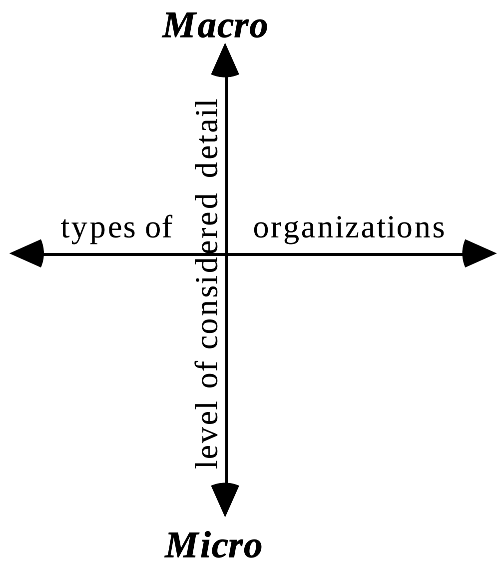

Figure 2. The two dimensions of considered organizations

The horizontal level differentiates between different organizational forms, such as hierarchical or matrix organizations. But also virtual organizations must be considered. The term virtual organization is used to describe organizations which exist mainly by the provision of electronic communication media, i.e. which share no common physical working environment such as office buildings. The working environment can be provided either by simple e-mail connections and shared file access between the participating sites or by a virtual reality system such as the systems proposed by strand 4 or a virtual office system such as DIVA (Berlage and Genau, 1993). Such systems aim to provide the organization members with a virtual organization environment similar to the physical one. For the support of such applications and their users, the OIS must be also capable to model the specifics of the virtual organizations world and offices

The vertical dimension differentiates between the levels of detail that is considered. At the macro level the organization is viewed as a whole and the basic building blocks and structures of the organization are considered. That level describes typically the view an external observer gets from the organization. The macro level could perhaps be suitably represented and published to the outside world via the X.500 Directory (X.500, 1993). If we take a deeper insight into an organization and go beyond the skeleton we reach the micro level. Here the organizational details of smaller entities is of interest, e.g. projects, research groups, committees. This results in a more detailed modelling of the organizational components and relationships and thus a finer granularity of access rights and administration rules is required.

Both dimension are clearly interrelated. Therefore an extendible and adaptable OIS organization model is required to capture all forms of organizations in the horizontal dimension as well as all levels of details in the vertical dimension.

The Conceptual Organization Model

The conceptual organization model identifies the organizational entities and the organizational relationships between them.

As organizational entities, those entities of an organization that determine and describe the working context of users are considered. Reflecting the different dimension of organizations introduced previously, these includes information about the employees, projects, roles, committees, departments, locations of an organization. Furthermore, the resources of cooperation such as documents, calendars and other kinds of commonly used data must be considered. On this modelling level the properties and behaviour of an organizational entity is not yet specified. This will be done later at the object model level.

However, all these discrete organizational entities become expressive only when they can be related to each other. Therefore we need ways to describe organizational relationships that place the organizational entities into a context: who is member or leader of a project, which projects are undertaken by a particular department, who is the projects secretary, who is occupying the role of the technical administrator of a special file-server, or who supports which task, or which forms do I need to apply for an organizational procedure? The combination of organizational entities by relationships allows the description of common tasks and procedures of the organization. For example, single task descriptions can be linked together by temporal our causal relations and each task description can be linked to a responsible role or a resource such as a document or form. These descriptions are useful as resources for the organization of a cooperative activity.

The examination of the related work in the field of organizational information systems (Prinz, 1995e) has shown that the approach of modelling an organization by the identification of its organizational entities and relationship is appropriate and useful. Semantic nets are a commonly applied technique for the description of the conceptual model. Another description technique is offered by the entity-relationship model (Chen, 1976). I will not discuss differences and advantages of both approaches as a potential description technique of the conceptual model here. The major aim of both approaches is the identification of the entities or objects of an organization and their relationships for further transformation into a data model. Accordingly, both description methods can be found frequently in object oriented analysis methods (Stein, 1993). For example, the Object-Oriented Software Specification method by (Bailin, 1989) uses entity relationship diagrams, others such as Object-Oriented Design by Booch (Booch, 1991) or the Object Modelling Technique by Rumbaugh (Rumbaugh et al., 1991) use description methods similar to semantic nets. Thus, from the viewpoint of our purpose their major difference lies in their representation methods. Therefore the meta-object model that is presented in the next section is capable of transforming both approaches into an organizational object model. However, a useful discussions can be found in (Carlson and Ram, 1992), (King, 1989), or (Hull and King, 1987).

The conceptual model consist of the identification of the organizational entities and relationships of a particular organization. It results from an analysis of the application domain. The next step is the specification of the properties and behaviour of each entity and relationship. Then a meta-object model is introduced that presents a framework for the specification of both aspects.

The next section presents an event distribution model that is based on the conceptual organization model and which aims to provide awareness about changes to the organizational information base.

An Event Distribution Model for the Provision of Awareness about Organizational Changes

General Considerations

The provision of awareness about organizational changes has been identified as a requirement for an organization information service. This section presents an event distribution model that provides mechanisms for increasing the awareness about organizational changes. It is based on a context dependent notification of users about modifications to organizational objects which concern their working environment. The model extends the initial ideas for the provision of awareness in cooperation support object system presented in (Mariani and Prinz, 1993). More general considerations about the model can be found in (Fuchs et al., 1994) or (Fuchs and Prinz, 1994) and a video that illustrates the application of this idea to a shared workspace application. In the following some general thoughts and requirements about an event distribution model will be presented, followed by the outline of an event distribution algorithm.

Information about changes in a cooperative working environment should be distributed in the context of the work in which they occurred. For example the notification about a new or modified document should be distributed to those people who are involved in the production or processing of that document, i.e. within the organizational boundary of that document. In the currently used systems information about such an action is often distributed explicitly by user produced meta-messages which contain a content like "I produced a new version of the project document". In order to release users from the explicit production of such meta-messages the event model should support an automatic generation of events caused by the processing of a work artefact (such as the document) as well as their distribution in the appropriate context, i.e. the organizational boundary.

Work artefacts can be part of several different working contexts and they can play different roles in each. For example, a document can be the major process artefact in a specific context and the same document can be a background artefact in another context. Thus, information about the processing of the document must be distributed differently in both contexts based on the different roles, i.e. the different semantic associations of the document.

For events that cross organizational boundaries it should be possible to modify, e.g. summarise or reduce the information content of the event. For example, the detailed information about a new document that might be delivered to the directly involved people, could be reduced to a short notification when it is delivered to an associated or cooperating group. That case occurs often when a project consists of small subgroups. Then the subgroup members want to be informed in detail about events in their group, while they might only want to receive a short notice about actions that happened in one of the other groups. Where adequate, the event distribution model should allow for an tailorable handling of that case.

The event information should be distributed and presented at the 'right' time to avoid disturbing the user who receives the notification. The right time depends on the importance of the event and on the current working focus of the user. For example, events that occurred in a working context are collected until the user moves to that context or until a certain number of events are collected.

From these considerations a set of requirements for a computational model can be derived.

Firstly, a representation of the work context and organizational boundaries that link or separate different work contexts is needed. This can be achieved with the organizational model presented in this chapter. Work artefacts, (e.g. documents, software) and the set of people who use or process these artefacts are represented by organizational objects of the appropriate classes. The role that a work artefact or member plays in a specific context is described by the relationship that links its object representation to a context description object such as a project, committee are workgroup object. Furthermore, organizational relationships that exist between the artefacts within a context (e.g. authorship) can be described with semantic relationships between the various elements. The following figure presents an example of such a semantic net.

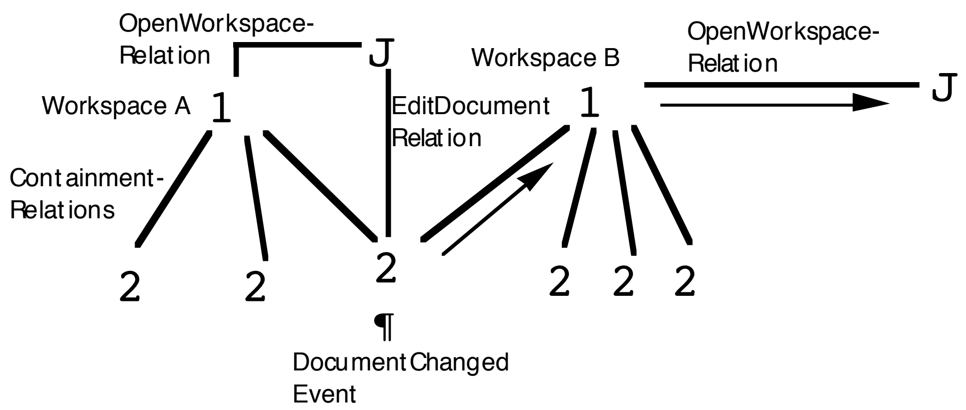

Figure 3. Example for the representation of a working context (Fuchs et al., 1995)

Secondly, the type of relationship by which an artefact is bound to a specific context determines how events raised by the processing of the artefact will be distributed. For example, if a document is a working document of a context, then all changes to the document will be notified to the context members. But, if the document is just a background document, then only new versions of the document may need to be notified. Thus, we can say that event distribution depends on the type of event that is raised by the processing (e.g. change, new version) and the type of the relationship that links the object to a context (e.g. working document, background material) So, we can say that the relationships are the medium by which events flow within a context. For some events, a relationship serves as a medium (working document relation for a change event) and for others it doesn't, i.e. it blocks the event (background material relations for a change event).

Associations between different work context, such as they exist between cooperating projects or subgroups of a project, can be described by organizational relationships too. Such specific relationships can be interpreted as the representation of an organizational boundary between two different contexts. Relationships that represent organizational boundaries perform specific operations on the events they transport. Adopting the media metaphor for that purpose, we can say that the media has a resistance. This media resistance leads to a modification of an events intensity. This reflects the real world observation, that information is often transformed when it is passed from one context into another. It furthermore bears the effect, that events are not distributed indefinitely. By reducing the events intensity whenever an organizational boundary is passed, the event fades out. That can be used for different reactions on an event and in the end it leads to a deletion of the event.

Assuming that an organizational model includes the definition of a large set of organizational relationships, the administration of the media resistance factors for each relationships appears to be very difficult. However, since an object-oriented model is applied to the modelling of organizational relationships this problem can be eliminated or reduced. Firstly, the resistance factor of a relationship is defined with its class definition and not independently for each instance. This reduces flexibility, but it is a prerequisite for the manageability of the approach. Second, inheritance allows the definition of resistance factor on an abstract level in the relationship class hierarchy. We will see in chapter 6 that the first level of the relationship class tree contains four different abstract classes: structural, procedural, resource, and spatial. If a resistance factor is defined only for each of these subtree-roots, the event distribution model becomes applicable. Further refinements are then possible by additional tuning of the subclasses, if necessary.

The next section presents the event distribution algorithm.

The event distribution algorithm

The following event distribution algorithm incorporates the consideration presented above:

I) Modifications on organization objects trigger an event object of the class that reflects the type of operation. This event carries the following information:
- a unique identifier, used to determine the origin of copied event objects.
- the distinguished name of the user who performed the action,
- the name of the object that is modified, i.e. that triggered the event
- the data that has been changed, e.g. the attribute name and its new value.
- a date/time stamp
- an initial default intensity level

II) The organization object that triggered the event delivers it to all connected relationship objects. If there is more than one relationship object, the event object is copied. The unique identifier is kept the same to allow for a later identification of copies of the original event object.

III) Each relationship object decides whether it is able to distribute the received event. This decision is based on the following conditions.

a) First, the relationship object checks whether it represents the right medium for the event distribution. If that is not the case, the event is dropped. That decision is based on the type of event. Each relationship object class specifies for that purpose a list of event type names, for which objects of the class can serve as a distribution medium. An event object can be forwarded when the name of its class or any of its superclasses is listed for the relationship object class. Otherwise it is dropped, i.e. deleted. This technique allows the specification of context sensitive event distribution channels.

b) Second, the relationship object checks whether the intensity of the event exceeds a certain threshold. This threshold can be regarded as the resistance of the relationship. If the events intensity is below the threshold it is not further distributed and dropped.

IV) If the event survives the third step, its intensity is reduced by the threshold of the relationship object. Then it is forwarded to all organization objects to which the relationship points. If there is more than one, copies of the event object with the same unique identifier are produced.

V) An organization object that receives an event reacts in different ways depending on its class.

a) If the organization object represents an actor i.e. an employee or a group, then the corresponding user is notified about that event. How the user is notified depends on the preferences specified in special attributes of his representing object.

The *notifyMedia* attribute determines the media that should be used for notification, e.g. email, immediate windowPopUp.

The *notifyThreshold* attribute determines the minimum intensity level of an event object that causes an immediate notification.

The *notifyCollect* attribute determines how many events that fall below the notifyThreshold should be collected until an aggregated notification is produced.

The *notifyPeriod* attribute determines the maximum time interval between two aggregated notifications.

Due to the fact that event objects are copied whenever more than one distribution path exists, it is possible that two event objects with the same unique identifier, i.e. the same origin are received. This will lead to a repeated user notification of equal event objects. Thus each receiving object holds the event object for a certain amount of time[21] until a notification is produced. If a second event during this time is received with the same unique identifier, then the event with the lower intensity level is dropped.

b) Other organization objects add the event content to a local event history list. Then the event object (and its copies) is propagated to connected relationship objects. The process continues with step III.

Two primary goals are achieved with the outlined event distribution model . Firstly, information about organizational changes is automatically distributed in the organizational context of the modified object. Secondly, the intensity dependent diffusion of event objects together with the automatic decreasing of its intensity allows the modelling of distances between the place where the action occurred and where it is received. This enables a fine grained reaction to events and ensures a finite distribution of event objects. A further advantage of this approach is that it ties in neatly with the introduced organization representation method, i.e. no further modelling techniques are required.

The following section presents a meta-object model that introduces a framework for the specification of the organizational object model, including the prerequisites for the application of the event distribution model presented in this section.

The Meta Object Model

This section presents the data model for the representation of the conceptual model. An important requirement for a toolkit-oriented organizational information service in a CSCW environment is the extendibility and adaptability of the organization model. Thus an object-oriented model has been chosen because it allows one to define a set of basic organization objects which can be specialised to meet specific organizational requirements. Furthermore, this approach enables the modelling of the various facets and the behaviour of an organization entity or relation to be focused in a single object class. This simplifies the creation and administration of

21 The length of this time interval depends on the distance between the object that originally triggered the event and the receiving object. If the originating object is located in a different domain (indicated by its name) to which only WAN connections exist then a longer time interval must be applied. Otherwise the time interval can be specified in the range of seconds. Nevertheless this, as many other variables of this model must be determined empirically.

the organization object model. Furthermore, it allows the definition of different organizational objects with a distinctive behaviour. The lack of this property has been identified as a weakness of the X.500 Directory modelling capabilities (Prinz, 1995d).

The meta object model specifies how the model elements for organizational modelling , i.e. the object classes of the organization object model are defined. Thus it specifies a framework for the organizational object model. This framework consist of four basic components:

I) the organizational object classes,
II) the organizational relation classes,
III) attributes,
IV) events and an event distribution model.

The first two components correspond to elements of the conceptual organization model. They form the basic skeleton of the organizational model. The attribute component contains the specification rules for basic attribute types which will be used to describe the properties of organizational objects and relations. Events and an appropriate event distribution model represent the fourth component. This model presents a mechanism for the provision of awareness about organizational changes, i.e. it fulfils the corresponding requirement for an OIS (Prinz, 1995e).

The following sections present the definition of each component of the meta-object model.

Definition Elements for Organizational Objectclasses

Organizational object classes are used to define a schema for the representation of organizational entities. Examples are object classes for the representation of employees, project descriptions, roles, committees, message types, rooms, buildings.

The meta-object model distinguishes between

- constructive,
- administrative, and
- descriptive

definition elements.

Constructive are those elements which are required for the construction of a workable and useful object class specification or instance creation respectively. **Administrative** elements contain specifications which support the management of the object model as well as instances of the specific object class. **Descriptive** elements are used to augment a class definition with additional information that is of further use for its application.

The elements for the definition of an for organization object class are illustrated in figure 4. Constructive elements are indicated by shadowed, administrative by grey, and descriptive elements by solid white boxes.

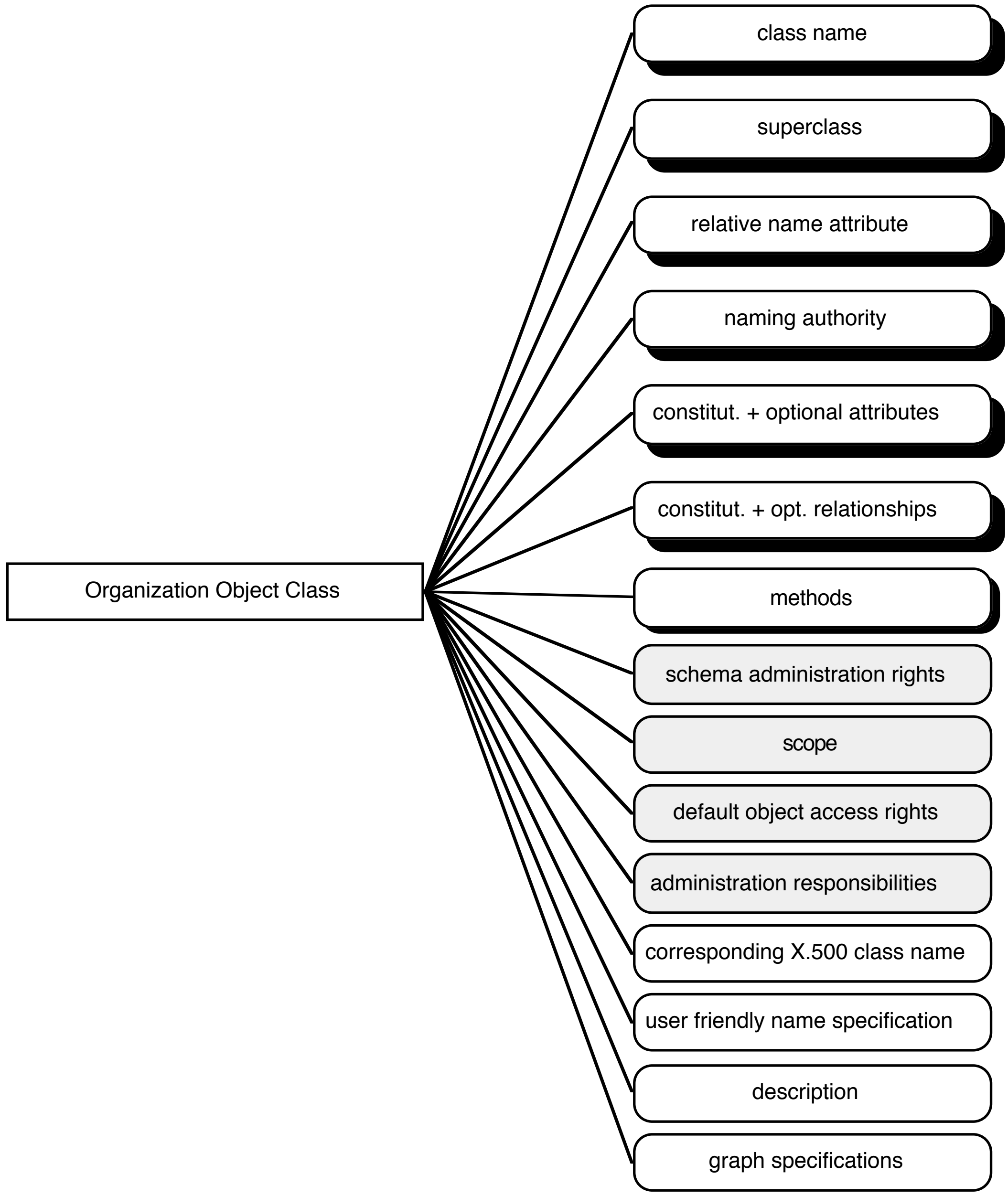

Figure 4. Elements of the definition of an organization object class.

The selection of the elements shown in figure 4 is determined by general object-oriented modelling considerations and the organization model requirements identified in (Prinz, 1995e). In the following sub-sections each model element is described in detail and the category the element belongs to is indicated.

For each of the following model elements an illustrative example will be given. It describes the appropriate assignments to the meta-object model elements for the *project* object class.

class name - *constructive*

The name is a unique identifier for the object class. It should be chosen in a way that it describes the semantic of the type.

Example:

```
class Project
```

super class - *constructive*

The object model supports single inheritance, so that for each class one super class must be identified. The root of the class hierarchy is a special object class which includes the basic functionality for all subclasses.

Example:

```
class Project subclass of Task Unit
```

The naming concept:

I) **the relative name -** *constructive*

II) **the naming authority -** *administrative*

The identification and retrieval of objects from a large object base requires that each object can be uniquely identified. It is a common approach in object systems to assign unique object identifiers. Often these are plain numeric identifiers. ONTOS (Ontos, 1992) for example uses 64 bit identifiers for that purpose. In contrast to that plain naming schema (Kühnhauser, 1991) examines several alternatives for naming concepts in distributed systems which are all based on a hierarchical naming schema. Since hierarchical naming schemas allow the separation of the name space of large object sets into small and administratable areas, this is the most appropriate naming approach.

For the provision of distributed management mechanisms the application of a hierarchical naming schema similar to the one that is specified in the X.500 Directory standard is appropriate. This requires that each object has assigned a **relative name**, which identifies the object uniquely in its local name space. The local name space is determined by the relative name of all objects at the same level in the name tree that belong to the same superior object. That superior object is called the **naming authority** for all objects which are a direct successor of the object in the name tree.

The name that identifies an object uniquely in the name space of the whole object base is called its distinguished name. The distinguished name of an object is constructed by concatenating all relative names of superior objects in the name tree, starting at the root. For example, the distinguished name of the object with the relative name "Institute FIT" is: "DE, GMD, Institute FIT“, if we assume that the name tree path for that object starts with the object called "DE", followed by the object called "GMD". The example illustrates the difference between the TOSCA and the X.500 Directory distinguished name syntax: The X.500 name syntax for the same object would appear as: "c = DE, o = GMD, ou = Institute FIT". Since each TOSCA object has its relative name represented by the same naming attribute, the name syntax does not contain an identification of the naming attribute, e.g. "c=", "o=", "ou=".

One key component of that naming concept is the existence of so called naming authorities. A naming authority[22] is an instance that is allowed to assign a relative name to an object that is created within its naming domain. The naming domain covers all objects which are direct successors in the name tree. The relative names of these objects construct the name space.

The information about adequate naming authorities for an object class is later used by administration interfaces to guarantee a consistent naming in an object system which is cooperatively administered. The naming authority information is provided by listing the names of object classes whose instances can be naming authorities to instances of the actually defined class. The property to be naming authority for a class is inherited by subclasses. This allows the use of classes which are defined at a 'high' position in the schema tree to avoid the necessity for listing all further subclasses which might also apply.

Example:

The naming authorities of the object class Project are Organization or Organizational Unit.

naming authority: `Organization, OrgUnit`

relative name: -- *to be assigned at object instance level*

The relative name is not defined at the class definition level. Its appearance in the meta-object model specifies that it has to be an element of each object class.

constitutional and optional attributes - *constructive*

The object model distinguishes between constitutional and optional attributes of an object class. Constitutional attributes are mandatory attributes which are essential for the characterisation of an object and which cannot be left empty when describing an instance of an object class. For example, the naming attributes, or communication addresses for object classes used to represent communication services. The information should be applied by an administration (user) interface to enforce the provision of values for these attributes. Optional attributes are those attributes which provide additional useful information, but which are not strongly required for the applicability of the object.

Example:

constitutional attributes: `projectNumber, phone, fax, email`

---

22 The concept of naming authorities is comaparable to X.500 DIT structure rules.

```
optional attributes: description, duration, meetingDates
```

constitutional and optional relationships - *constructive*

Organizational relationships between instances of organizational objects classes are represented by objects of a special relationship object class (see following section). Like attributes, relationships can be either constitutional or optional. Accordingly, for each object class a list of constitutional and optional relationships must be supplied. Often, constitutional relationships are relationships which link an object into its organizational context.

The information about constitutional and optional attributes or relationships is used to ensure schema correctness of the information base.

Example:

```
constitutional relationships: projectOf, projectMembers,
                              projectLeader
```

The `projectOf` relationship describes the organizational context of the project by linking its entry with an organization or organizational unit entry. The `projectMember`, and `projectLeader` relationships point to individuals, e.g. employee entries.

```
optional relationships: relatedProjects, resources,
                        producedSoftware
```

The optional relationships describe associations with other projects, the resources required by the projects, the software that has been produced by the project, or the timetable for project meetings.

methods - *constructive*

The behaviour of an object class is defined by its methods which define the services that objects of that class will offer to other classes. Each object class inherits a set of basic methods from a basic organizational object class which builds the root for the whole object class schema. These methods provide the necessary functionality to create, maintain and access the objects, their properties, and relationships. A description of these methods is presented in chapter 6.

The following example illustrates the definition of additional methods to describe the special behaviour of the project object class.

Example:

```
methods: inviteNextProjectMeeting(message)
         list  listMembers()
         list  listMembersEmail()
         ...
```

The `inviteNextProjectMeeting(message)` method notifies all project member about the next project meeting using `message` as the message text.

The `listMembers()` methods returns all members of the project by evaluation of the `projectMembers` relationship. The functionality of that method can be also achieved by an appropriate application of the basic method `getRelObj(relationIdentifier)` (chapter 6) that returns all objects which are related by the `relationIdentifer` relationship (e.g. `projectMembers`). However, providing that functionality by a specialised method provides a more convenient access. Another method of that kind is the `listMemberEmail()` method which returns the email addresses of all members by reading the `email` attribute value from all objects returned by the `listMembers()` method.

schema administration rights - *administrative*

To control modifications to the object class schema, each object class defines schema administration rights. They determine:

- who is allowed to modify the schema of this class (`modifySchema`)– this includes modification to the attributes and methods as well as modifications to all other elements of the meta-object model, e.g. *scope* or *user friendly name specification.*
- who is allowed to create subclasses of this class (`createSubclass`)

These access rights are enforced by the object model designer (Prinz, 1995c).

Example:

```
modifySchema: (DE, GMD) -> TOSCAAdministrator

createSubclass: (DE, GMD) -> institutes -> TOSCAAdministrator
```

This example describes that the schema for that class can be modified only by the TOSCA administrator of the organization (GMD). However, subclasses to that class can be created by the different administrators of the institutes of the organization. For the project class example, this specification allows each institute to extend the basic project object class by institute specific needs.

scope - *administrative*

On the micro level of organization modelling an OIS is used by small groups. The members of projects or working groups model their own organizational context information, like distribution lists, communication addresses, message forwarding, documents, tools. The requirements for the definition of the attributes, relationships, and services of an object class might differ between different groups, application contexts, or the general organization definition. This requires a concept which allows the definition of new or specialised object classes which are valid only for a specific organizational context.

This can be achieved by the assignment of a scope specification to an object class definition. This can be interpreted as an access right on the schema level. A scope specification describes the set of objects which have access to the object class schema description as well as to instances of this class. Usually these are objects which represent actors in the systems, e.g. members of the project for which this new object class is defined. The scope specification is described using the object retrieval language described in (Prinz, 1995b). Thus, the scope specification of an object class definition contains an expression in that language. The evaluation result of this expression is interpreted as the set of objects which represent actors who have access to that object class and its instances.

The scope element of the meta-object definition is comparable to the X.500 Directory definition of subschemas for administrative authority domains (Prinz, 1995d). The difference of the method presented here is that it does not depend on the hierarchical organization of the information objects, because the scope can be defined with a query language that allows specifications which cross organizational information hierarchies.

A problem with the scoping concept is the proliferation of object classes with similar semantics in different context. This bears the risk of a wild growing number of object classes in the long term. This also avoids or restricts the sharing and exchange of objects between different organizational contexts. However, I believe that the freedom of user groups to choose their own representation of work context instead of getting it dictated from a general schema (Suchmann, 1993) is a good argument for that approach. Nevertheless the application of the scoping concept requires user discipline. Approaches for the assimilation of different object class schema trees can be found in (Lee and Malone, 1990). Complementary to that approach the EGRET framework (Johnson, 1992) supports the identification of classes with similar semantics in a cooperatively used object system.

Example:

```
scope:     (DE, GMD, *) alternative (DE, GMD) -> employees

scope:     (DE, GMD, FIT, COMIC) -> project_members
```

The first scope definition specifies that the object class is valid for all definitions within the scope of the organization GMD.

The second example specifies the scope of an object class that can be used by the members of the COMIC project, assuming that the name (..., COMIC) denotes a project entry.

default object access rights - *administrative*

For the simplification of access right specifications on the object level, each object class definition may contain the description of default access rights. These access rights apply to all attributes of an object class unless they are overwritten by specific access rights for the attributes or relationships of a class.

Three types of access rights are distinguished:

- creation of a new object[23]
- modification access
- read access

[23] This access right makes sense on the object class level only.

For the specification of access rights, three major methods can be distinguished (Landwehr, 1981):

- access control lists,
- capabilities, and
- clearences and classifications.

Access control lists and capabilities are derivations from the access matrix model that was developed in the early seventies. An access control matrix contains one row for each subject (e.g. a user) that accesses an entity and one column for each object (e.g. a data entity or a service) that can be accessed by a subject. A matrix entry specifies the access rights of a subject to an object. Since all possible subjects do not access all possible objects, many fields in the access matrix will remain empty. It is therefore sensible to decompose the matrix row by row or column by column thus eliminating the empty entries. That decomposition leads either to access control lists or capabilities.

Capabilities are associated with the accessing subjects. They indicate for an object the operations the subject is allowed to perform. Within implementations, e.g. in the AMOEBA system (Mullender and Tanenbaum, 1986), capabilities are often represented by bit-strings which can be regarded as an access key to an object. This key is examined by an accessed object to check whether access shall be denied or allowed.

Access control lists are associated with objects. They indicate the subjects and operations they are allowed to perform on the object. An accessing subject provides its identification to the accessed object. Then, the accessed object examines whether the provided identification (e.g. the subjects name) appears in the access control list for the intended operation. An example for an access control list application is the description of access rights at the UNIX file and directory level.

The clearences and classifications model introduced by Bell and LaPadula (Bell and LaPadula, 1973) is based on a classification of subjects and objects into security classes. Subjects are classified by being assigned a clearance, objects are assigned a classification. Bell and LaPadula define two axioms which determine the permitted access of a subject to an object:

a) No user may read information classified above his clearence level ("no read up")

b) No user may lower the classification of information ("no read down")

In (Prinz, 1992) these three models have been reviewed for their applicability to store authorisation information in the X.500 Directory. The advantage of the Bell LaPadula model is its simplicity, however its low granularity is of a disadvantage. Capabilities are well suited for an environment with a low fluctuation of objects and a high fluctuation of subjects. The opposite is true for access control lists. In an OIS we can expect frequent changes to subjects, i.e. the OIS users, and to the objects, i.e. the OIS data objects. For that reason both methods are applicable. However, the fact that access control lists are associated with the protected objects, enables their administration in the context of the object base, i.e. together with object they protect. This allows their integrated storage, evaluation and administration within the OIS. For these reasons the access control model is chosen for the specification of access rights in TOSCA.

The object query language is used for the specification of access control lists in TOSCA. Modifications to these access right specifications can be made by the user who owns the **modifySchema** access right (see above).

Example:

```
defaultAccessRights:

  create: (DE, GMD) -> institutes -> TOSCAAdministrator

  modify: this -> project_leader | this -> project_members

  read:   -- no specification: access granted to all
```

The default **create** right is assigned to the administrators of the different institutes of the organization.

With the default **modify** right we want to grant access to the project leader and members of a particular project instance. To achieve that at the abstract level of the object class definition we need an element that refers to the object instance of the object class. This element is provided by the `this` keyword, which is interpreted as the object instance, when the object query specification is evaluated. Thus, we are able to include object instance references at the abstract modelling level. The example shows how to grant the modify access right to the project leader and members which will later be related to the particular project object.

The default **read** access right is not specified in the example. This is interpreted as allowing everybody read access to the project object per default.

the administration responsibility - *administrative*

This element contains a default specification of the administrator who is responsible for the correctness and actuality of the objects and their attributes of that object class. A default description can be overwritten either on the particular object or the attribute definition level. This information can be used by administration services to identify and notify the responsible administrator. Chapter 7 demonstrates how that information is used to support a coordinated administration management.

Comparable to the specification of default access rights, responsibilities must be described in an abstract manner. For example, the general object class Device should contain a description that identifies the administrator role of the department to which a particular device object belongs as the actor who is responsible for the administration of the device object. The object retrieval language is also used for the specification of administration responsibility expressions. Actors who are contained in the responsibility description are also granted the modification access right, without being mentioned there explicitly.

Example:

```
administrationResponsibility: this -> project_leader
```

The responsibility for the correctness of the project entry is assigned by default to the project leader. We can see the difference between the **modify** right (see above) and the administration responsibility. The modify right is assigned to several users (e.g. the project leader and members) while the administration responsibility should be assigned only to a single user. This allows more specific notifications to the responsible administrator in cases where inconsistencies are detected or simply when questions about the entry need to be asked. This single point of responsibility is also useful for the automation of administrative procedures, because a multiplication of responsibility alternatives increases the complexity of the administration process.

the corresponding X.500 object class name - *descriptive*

The OIS requirement list (Prinz, 1995e) contains the request for information access to the X.500 Directory. On the object schema level this requires instructions for the mapping of objects retrieved from the X.500 Directory onto objects of the OIS. For that purpose the corresponding X.500 object class name can be provided for each TOSCA object class.

This information is used to map X.500 entries which have been retrieved from the Directory onto the appropriate object class in the organizational data model. With that technique a transparent mapping and presentation of X.500 information in TOSCA can be achieved. In (Prinz, 1995b) a concept is presented how the TOSCA information can be exported to the X.500 world applying the same information.

Example:

```
X.500objectClass: organizationalPerson
```

For the project object class we cannot define a corresponding X.500 object class, since that is not defined by the standard (X.521, 1993). Therefore we use the employee object class, which contains "organizationalPerson" as the corresponding X.500 object class name to illustrate that meta-object model element.

the user friendly name specification - *descriptive*

The distinguished name of an object is often not expressive or user friendly enough for its presentation in a user interface. Therefore, the user friendly name specification allows the specification of construction rules by which the name of an object can be constructed. The name can be built by a combination of attributes of the object as well as by retrieving information from objects where this object contains relationships to.

User friendly names are specified in the language developed for the specification of virtual attributes combined with the object retrieval language.

Example:

```
userFriendlyName: acronym + ", " + longname
```

The user friendly name of objects of the project object class can be built by concatenating the value of the attribute `acronym` followed by a colon and the value of the `longname` attribute.

```
userFriendlyName: givenname + " " + surname + ", "
                               ufn of this -> occupies_role
```

In this example the user friendly name of an employee object is constructed by concatenating the value of the `givenname` and `surname` attributes followed by a colon and by getting the user friendly name (`ufn)` of the Role object that this object is related to by the "`occupiesRole`" relation.

description - *descriptive*

The description element of the object class definition is used to provide a description of main characteristics and usage of the class. In large distributed system this is useful for both administrators and users who retrieve an object of an unknown type. Very often the presentation of the pure class schema is not sufficient for an understanding of the semantics of the class of the retrieved object.

Example:

```
description: "The project object class models descriptive,
             formal, task, and membership oriented aspects
             of GMD projects."

description: "The 'external project' is a subclass
             of the standard project object class. It contains
             additional properties to represent the external
             project partners and to support the scheduling of
             project meetings."
```

Two examples for the description of a standard project classes and a refined subclass.

graph description - *descriptive*

To supply a description of the context in which an object is embedded it is very helpful to provide a graphical view. Since the context is different for each object class, this view can not be generalised. The graph description element allows the specification of graph layout and construction rules for each object class. (Prinz, 1995a) presents how that information is used by the organization browser to display organizational charts which provide a graphical representation of the embedding organizational context of a focused object.

Example:

```
graphSpecification:
{
layout: Vertical

superiors:
```

```
[this];[this -> project_of];superior organizational unit

subordinates:

[this];[this -> project_members];project members
[this];[this -> project_leader];project leader
[this -> project_of];[this -> has_project];
                         other projects of the superior unit
}
```

This example demonstrates the specification of the graph for project objects. The description contains three sections. First, the layout direction of the graph is specified. The second and third section specify the objects that shall be displayed as superior or subordinate objects to the current object (denoted by `this`) in the graph using the object retrieval language. Each entry in these section consists of three parts. The first part specifies the source objects and the second part the destination objects for an arc of the graph. The third part is used as a comment.

The example describes a graph that displays entries for the superior organizational unit of the current project, its members and the project leader, as well as all other projects of the superior organizational unit.

Definition Elements for Organizational Relation Classes

Organizational relationship classes are used to define a schema for the representation of organizational relationships. This section describes the definition elements of the meta-object model of relationship classes.

In the literature two major alternatives are discussed for an appropriate modelling of relationships in object oriented systems . The modelling of a relationship as an attribute of the object is favoured in (Booch, 1991) or (X.500, 1993) while the alternative of modelling relationships as independent objects is introduced by (Rumbaugh, 1987) and later extended in (Rumbaugh et al., 1991), or (Bratsberg, 1991) and (Diaz and Gray, 1991) for example. This discussion will not be deepened here further (see (Frank, 1993) for more details). I believe that both approaches have their justification depending on the particular application scenario.

For our system the approach to model relationships as independent objects has been chosen for the following reasons:

- Relationships between organization objects are organizational facts, which can be categorised and classified. Thus it is appropriate to model their semantics as particular objects.
- Organizational relationships posses different facets, which can be suitable modelled as the properties of an object (e.g. the lifetime of a project member relationships can be expressed by an attribute).
- Modelling relationships as attributes would spread the relationship semantics over the organizational objects. Modelling them as objects keeps the entire relationship semantic in one place.
- The modelling of relationships as organization object attributes requires a class modification when new relationships are identified, followed by a object

reorganization. Since this is expected to occur frequently, the modelling of relationships as objects which are just linked to organization objects is more appropriate for a dynamic environment.

- The modelling of relationships by objects allows the definition of reusable libraries, similar to the organization objects.

Following that decision, for each real-world relationship a corresponding object type must be defined. Examples are relationship classes for the representation of structural relationships (e.g. department of, project of) or organizational relationships (e.g. associated with, cooperates with), or location relationships (e.g. located at, contained in), etc.

The elements for the definition of an organization object class are illustrated in figure 5.

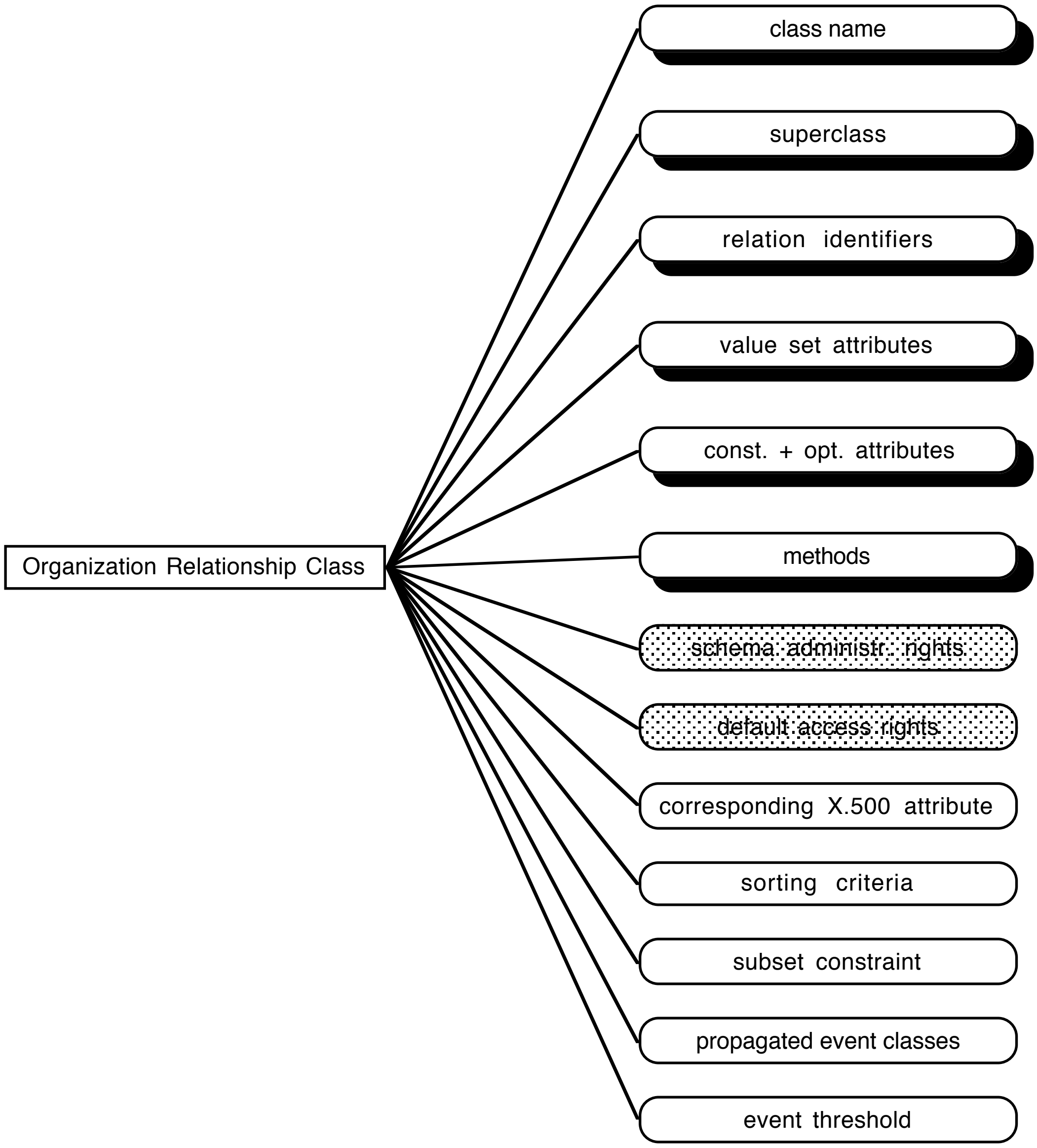

Figure 5. Elements of the definition of an organization relationship class

Again, as with the organization object classes the relationship object model distinguishes between *constructive*, *administrative*, and *descriptive* elements. In the following each model element is described in detail whereas it indicated to which category the element belongs. An example based on the project membership relation will be given for the definition elements.

class name - *constructive*

super class - *constructive*

Same as for the organizational object class.

Example:

```
relation class ProjectMembership subclass of Membership
```

the relation identifiers - *constructive*

A relation object describes a relationship between two real world entities, which are themselves represented as organization objects. The relation identifiers are the names for the two directions of the relationship. The are used as identifiers for traversing the relationship from one organizational object to the related organizational object. Thus it is possible to traverse a relationship without explicit provision of a relationship name. Furthermore, relationship objects will be accessed only via their associated organization objects since that object determines the context for the relationship, or via their membership to a class. Therefore it is not necessary to assign each relationship object a name, as it is necessary for the organizational object. However, since the relationship identifiers are used to distinguish a relationship object from others that are associated with an organizational object, e.g. by the object retrieval language, all relation identifiers must be unique. They should also not conflict with an attribute name.

The domain for each side of the relationship is defined by listing the names of organization object classes with the relationship identifiers. The instantiation of a relationship object is then allowed only between objects of the defined organization object classes. This definition may conflict with the definition of the *constitutional and optional relationship* element of the organizational object class definition. In such cases the relation identifier element of the relationship class definition has priority over the organization object class definition. The organization object model designer notifies the administrator if such conflicts or inconsistencies are detected.

The following example illustrates the definition of the relationship identifiers for a class that models a project membership.

Example:

```
src: {Project}, project_members

dst: {Employee}, project_member_of
```

The ProjectMembership relation object class can be defined between the object of the organizational object classes *Project* and *Employee*. The identifiers of the project membership relation link between a project and an employee object is called "project members" from the project view, but "project member of" from the employees view. These identifiers for a relation link are defined as the source and destination identifiers. They are further applied as identifiers in the object retrieval language.

The following figure shows a graphical representation of the ProjectMembership notation that corresponds to the notation[24] proposed by Rumbaugh (Rumbaugh et al., 1991, pg. 34).

---

24 A *relation identifier* corresponds to a *role* in the Rumbaugh notation.

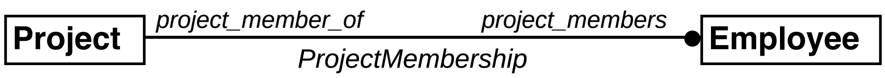

Figure 6. example for the graphical notation of an organizational relationship

value set attributes - *constructive*

The specification of the related organization objects which are linked by the relationship object is stored in two attributes: source-value and destination-value. The source-value is single valued while the destination value might reference a set of objects, i.e. the representation of 1:n relations is supported. For complexity reasons m:n relations are not supported. Experiments with the latter type of relationships in TOSCA have shown that they are difficult to understand both by users and administrators. Furthermore m:n relationships are more fragile in respect to organizational changes. If, for example, a new instance must be included at right hand side of the relation, for which the left hand side is no longer valid, the relationship has to be split in two new objects which requires additional administration effort. Although it will lead to an increased number of relationship instances it is therefore better to model an m:n relationships by the appropriate (m) numbers of 1:n relationships.

The attributes contain an expression formulated in the object retrieval language. This language allows the specification of objects either by naming or by expressions which allow for a dynamic description of the objects which are involved in a relationship. These expressions allow the description of organizational rules such as: voting members of this committee are the project leaders of all projects of the department. Thus it is not necessary to list all these people explicitly as it would be required in X.500. Also, they can be used to reduce redundancy by describing rules such as: employees of this institute are the members of all projects of this institute. Furthermore it is possible to define user dependent rules. This is needed for example, when the person who is responsible for a task, depends on the users membership in a project. In this case the actual user identity is needed to answer a request. For example, this is useful to support a role-based addressing of messages.

To ensure the unary cardinality of the source value, its specification must contain a name, i.e. no expressions are allowed which might denote more than one object.

Although the terms source and destination imply a direction, relations are not necessarily considered as directed.

Example:

```
srcValue: (DE, GMD, FIT, COMIC)

dstValue: (DE, GMD, FIT, Prinz Wolfgang),
          (DE, GMD, FIT, Fuchs Ludwin)
          (DE, GMD, FIT, OODB-Design) -> project_members
```

This example describes the project members of the COMIC project, i.e. a specific object instance, as two individuals and includes all project members of the OODB design project.

constitutional or optional attributes - *constructive*

If required, additional attributes for a further specification of a context dependent restriction of a relation can be defined. These can be used to model temporal relationships, i.e. relationships that exist only for a certain time interval (e.g. project or committee membership, substitutes).

Example:

```
constitutional attributes:    duration

optional attributes:       involvement
```

For the project membership relation the duration of a membership is defined as a constitutional attribute, while the involvement, expressed in a percentage for example, is considered optional.

methods - *constructive*

The root object class for all relationship objects provides a set of methods for the administration and processing of relationship objects. Additional methods can be defined to extend the behaviour of a relationship, i.e. to express temporal constraints on its validity, or to react on specific organizational circumstances or conditions. The appropriate conditions are then defined as additional attributes of a relationship object class.

Example:

```
methods:  isValid(date)
```

This is a method of a relation object class that evaluates a relationship attribute which specifies a condition for the validity of an actual relationship object. This may be a time interval which specifies the validity of a role-assignment, a substitute, or a membership (see `duration` attribute in previous example). This allows the expression of organizational relationships which are dynamic over time.

duration of an association and a research group This methods of a membership relation object class returns all project members who are involved in the project at "`date`". This method is useful when members of a project are part time workers or when employees are involved in more than one project.

schema administration rights - *administrative*

Same as for the organization object class.

default access rights - *administrative*

Similar to the access rights on organizational objects, default access and modification rights can be specified on relationship objects.

the corresponding X.500 attribute name - *descriptive*

For each relation object class that corresponds to a X.500 attribute, the X.500 attribute name can be provided. The X.500 attribute should have a distinguished name syntax, which allows it to be interpreted as a reference attribute. Then the values of that attribute point to another X.500 entry and they can be used to create a temporary relationship object upon retrieval. This mapping takes place when an X.500 entry is retrieved for which a mapping onto an organization object class is defined and when this entry furthermore contains attributes which correspond to a relationship object class.

Example:

```
X.500attribeType:      members
```

The members attribute type is defined in the X.500 Directory schema (X.520, 1993) with a distinguished name syntax. If a project entry is read from the X.500 Directory which contains this attribute, then its content is mapped onto an instance of the ProjectMembership object class.

sorting criteria - *descriptive*

A sorting criteria for the output of the organization objects specified by the dst value set attribute can be specified. This definition element defines a default behaviour for all instances of a relation type. User interfaces might specify other sorting criteria when they access relations.

Example:

```
sorting  criteria: surname
```

The project membership relation specifies that all employee objects should be sorted by the surname on retrieval. For other relations it might relevant to group objects by types (tools of a room), or by a location attribute (member organizations of an international project).

subset constraint - *descriptive*

Organizational rules often enforce that relationships between organizational entities may only exist if a required organizational context is fulfilled. For example, the chair of a committee must also be a member of the committee. Such organizational rules for relationships can be expressed by the subset constraint of a relationship class definition. It lists all organizational relationship classes to which a subset association exists.

Example:

```
subsetConstraint: projectMember
```

This example specifies for a project leader relationship that the related person must also be a project member of the project.

The two following elements are not directly related to the modelling of organizational relationships. They are required by the event distribution model that was introduced earlier.

forwarded event object classes - *descriptive*

This elements lists all event object classes for which objects of the relationship class can serve as a distribution medium, i.e. which the relationship provides a transport channel. Objects of the class will only forward events which belong to the listed classes. Events of all other classes will be ignored.

Example:

```
forwardedEventClasses:    Create, Modify
```

This relationship transports only creation or modification events. In the context of a project description it forwards events which indicate that either a new project member has been included to the project description or the project or an employees description has been modified.

event threshold/resistance - *descriptive*

The event threshold is specified by a number in the interval [0,1]. It determines the minimum intensity of an event object to be forwarded further. It is also the argument by which the event objects intensity is reduced before being forwarded.

Example:

```
threshold:    0.2
```

This very low level indicates that the project membership relation is a good transport media. This is caused by the fact that this relationship expresses an important and close relationship within an organizational context.

Remarks:

A relationship object class definition does not contain an **administration responsibility** specification, because that can not be defined sufficiently on the abstract class level. Instead the administration responsibility of a relationship object instance is derived from the responsibility of the organization object that is associated with the source link.

Attributes

Attribute Definition

The properties of an organizational object or relationship class are expressed by its attributes. Attributes are defined independent of an object class. Thus the semantic and definition of an attribute, (e.g. e-mail address) is the same for all its instances independent of the object class in which it is included. Thus applications can deal with a certain attribute of an object independent of the specific object class. This understanding of attributes can also be found in the X.500 Directory standard (X.520, 1993)

The elements of an attribute definition are illustrated in figure 7.

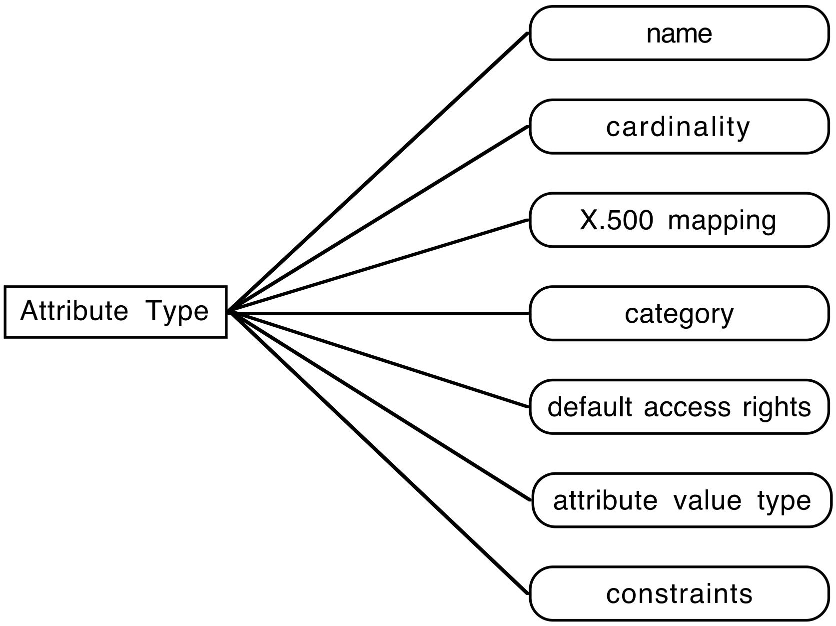

Figure 7. Elements of an attribute type definition

The different elements are now described in turn.

name

The domain of attribute names is the whole object schema, i.e. an attribute is named independent of an object class in which it is included. The name should express the semantics of the attribute, e.g. phoneNumber, emailAddress, description, accountNumber, duration.

cardinality

The cardinality of an attribute is either single or multiple.

X.500 mapping

When an entry of the X.500 Directory is retrieved then it is checked whether its class can be mapped onto an OIS organizational object class applying the X.500objectClass element of the class definition. Then, the attributes of the Directory entry are examined if they can be mapped onto an OIS attribute type using this definition element.

category

An attribute can be categorised as *simple, indexed,* or *naming*. The default category of an attribute is simple. For indexed attributes a special index is created that allows quick retrieval. Naming attributes are used to store the name of an object in addition to its relative name. These attributes are examined for access operations which do not specify a distinguished name. The values of the naming attributes must not be unique. This allows a less unspecified access of objects, however a simple read access might retrieve more than one object. Naming attributes are indexed by default.

Acronym is an example for an attribute type that is specified as *naming*. This attribute type is useful to describe entities such as projects or organizational units. They often carry an acronym together with their long name, e.g. ORGWIS stands for the "Organizationswissensbasis" project. By specifying the acronym attribute as naming, the project entry can be accessed either by the query `name = "COMIC"` or `name = "Computer based Mechanisms for Cooperative Work"`.

However, naming attributes are examined only by a non-distinguished name query, such as `name = "..."`. Queries specified in the distinguished name syntax, e.g. (DE, GMD, FIT, `COMIC)` do not examine naming attributes to ensure the retrieval of a unique entry.

default access rights

The specification of a default access right for on attribute overwrites the default access rights specified on the object class level. An attribute default access right can be overwritten at the object instance level. The object retrieval language is used for the specification of administration responsibility expressions.

administration responsibility

The object class definition contains a general specification of the administration responsibility for objects of a class. The specification is valid for all attributes of a class unless it is overwritten by an administration responsibility specification on this attribute definition level.

It is foreseen that most attribute definitions will not specify this element. It will be used mainly for the specification of responsibilities for technical (e.g. service specifications), address (e.g. mail, phone), or organization administrative attributes (e.g. project account numbers).

The object retrieval language is used for the specification of administration responsibility expressions.

attribute value type

The value type can be selected from the following list: Numeric, String, Virtual, Computational, Command. Their semantics are described in chapter 6.

constraints

Constraints can be specified to restrict the values of an attribute, e.g. salary, competences. It is assumed that constraints are expressed at the modelling level by a natural language description which is implemented by the corresponding modify methods of the attribute class.

The following section will now describe the definition elements of an event object class.

Event Objectclasses

The elements of the event object class definition can be derived straightforward from the event distribution mechanism outlined earlier.

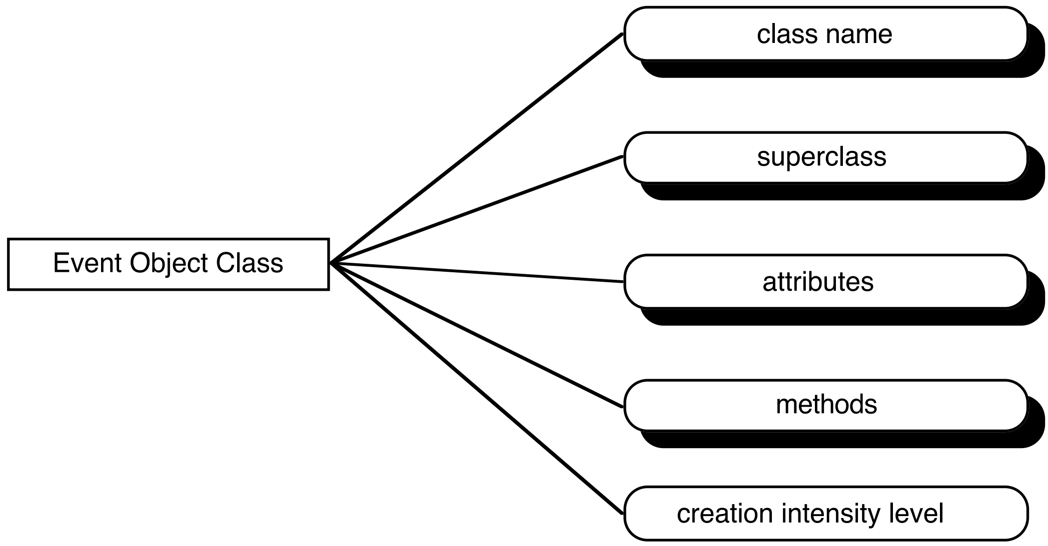

Figure 8. Definition elements of an event object class

Figure 8 presents an overview of the definition elements which are explained in detail in the following.

object class name - *constructive*

super class - *constructive*

Same as for the object class definitions in the previous sections.

attributes - *constructive*

The distribution algorithm requires a set of attributes which describe the origin of an event object and the actor and operations which raised the event. Furthermore the actual intensity level must be stored. These mandatory attributes are provided by a basic event class. Subclasses might specify further attributes which store additional information such as details about the data that has been modified.

methods - *constructive*

Each event object provides methods to replicate itself and to reduce the actual intensity level. They are provided by a basic event class. Further application specific methods may be included.

creation intensity level- *descriptive*

At creation each event object receives a class specific initial intensity level. This is specified here as a number in the interval [0,1].

Summary

This chapter presents the framework for the modelling of organizational information with TOSCA. The kernel of the framework is the meta object model which defines the construction rules for the creation of the specific object model that represents a particular organization. The following table correlates the requirements identified in (Prinz, 1995e) with the properties of the meta object model.

Representation and provision of organizational structures, entities, and relationships	The explicit distinction between organizational object and organizational relation classes allows an expressive modelling of organizational objects and in particular organizational relationships. This allows both, a modelling of (hierarchical) organizational structures as the skeleton of the organization and the modelling of more dynamic cooperation oriented relationships.
Provision of information about organizational services and procedures	The meta-object model provides the necessary tool-box for the modelling of services by organizational objects and their interdependencies by organizational relationships. The concrete application of this tool-box will be demonstrated in chapter 6 and the appropriate provision to users and integration of the service information into applications is demonstrated in chapter 7,8,9.

Communication support information	This is mainly satisfied by the definition of appropriate attributes for the various object classes and the extendibility of the model which allows it to react on the appearance of new communication addresses (e.g. WWW - URLs). However, the attribute value type *computational* and *virtual* allow for an inheritance, aggregation or computation of address subparts and sequences. This supports the administration of address information.
Provision of awareness about organizational changes	The presented event distribution model presented satisfies this requirement. It provides the mechanisms for a context based distribution of event objects which indicate changes to the organizational information base.
Data security and access control	Access rights can be defined at the schema and object level of the organization model. This enables access control to the organizational schema and data.
Support for distributed management of the organizational information	The administration responsibility elements of the organizational object and relation classes allow the assignment of administration responsibilities to the different administrators in a distributed environment. This is further supported by the adaptation of the hierarchical naming schema from X.500.
Flexible and extendible organization model	This is mainly achieved by the application of the object oriented inheritance mechanism and it is supported by the provision of an organization object model designer introduced in (Prinz, 1995c).
Integration with the X.500 Directory	The organizational object and relation classes contain both an element which specifies the mapping between the TOSCA and the X.500 object model. (Prinz, 1995b) describes the usage of these model elements to integrate X.500 Directory information.
Integration of external applications and resources	This is supported by the command attribute value type of the attribute class.

Architectural issues	The consideration of schema administration rights and administration responsibilities at the meta-object level separates the organizational distributed administration from the technical distribution. This enables an independent technical distribution without the consideration of organizational constraints.

This table shows that most of the requirements are captured by the organizational modelling framework of TOSCA. The framework provides the means for the extended organizational information modelling together with means for the modelling of security, administration, and X.500 integration aspects. Furthermore the proposed event distribution model incorporates the aspect of awareness about organizational changes. Those requirements which cannot be completely satisfied here, will be considered later in one of the following chapters as indicated in the table.

The following chapter presents how the framework introduced in this chapter is applied to define a basic set of organizational object classes for TOSCA.

References

Bailin, S. C. (1989): An Objects-Oriented Requirements Specification Method. *Communications of the ACM*, vol. 32, no. 5.

Bell, D. E. and L. J. LaPadula (1973): *Secure Computer Systems: Mathematical Foundations*. ESD/AFSC.

Berlage, Thomas and Andreas Genau (1993). *ACM Symposium on User Interface Software and Technology, Atlanta, GA*.

Booch, G. (1991): *Object-oriented Design with Applications*. Redwood: Benjamin Cummings.

Bratsberg, Svein Erik (1991): FOOD: Supporting Explicit Relations in a Fully Object-Oriented Database *Object-Oriented Databases: Analysis, Design & Construction*Elsevier Science Publishers B.V. (North-Holland), pp. 123-139.

Carlson, David A. and Sudha Ram (1992): A Knowledge Representation for Modeling Organizational Productivity. *Journal of Organizational Computing*, vol. 2, no. 2, pp. 155-180.

Chen, P. P. (1976): The Entity-Relationship Model: toward a unified view of data. *ACM Transactions on Database Systems*, vol. 1, no. 1, pp. 9-36.

Diaz, Oscar and Peter M. D. Gray (1991): Semantic-rich User-defined Relationships as a Main Constructor in Object-Oriented Database *Object-Oriented Databases: Analysis, Design & Construction*Elsevier Science Publishers B.V. (North-Holland), pp. 207-224.

Frank, Ulrich (1993): *A Comparison of two outstanding Methodologies for Object-Oriented Design*. Arbeitspapier . GMD.

Fuchs, Ludwin, Uta Pankoke-Babatz, and Wolfgang Prinz (1994). *D-CSCW ´94 - Einführung von CSCW-Systemen in Organisationen, Marburg*, pp. to appear.

Fuchs, Ludwin, Uta Pankoke-Babatz, and Wolfgang Prinz (1995). *ECSCW '95, Stockholm*. Kluwer.

Fuchs, L. and W. Prinz (1994): Organizational Context, Events and Awareness. In J. Bowers (ed.): *A Conceptual Framework for Describing Organizations*. Lancaster: Lancaster University, pp. 173-192.

Hull, Richard and Roger King (1987): Semantic Data Modeling: Survey, Applications, and Research Issues. *ACM Computing Surveys*, vol. 19, no. 3, September, pp. 201-260.

Johnson, Philip (1992). *CSCW ´92, Toronto, Canada*. ACM, pp. 298-305.

King, Roger (1989): My Cat is Object-Oriented. In W. Kim and F. Lochovsky (eds.): *Object-Oriented Concpets, Databases, and Applications*. New York, NY: Addison-Wesley, pp. 23-30.

Kühnhauser, Winfried E. (1991): *Namensräume in verteilten Systemen*. GMD-Berichte. München; Wien: Oldenbourg.

Landwehr, Carl E. (1981): Formal models for computing security. *ACM Computing Surveys*, vol. 13, no. 3, pp. 247-278.

Lee, J. and Thomas Malone (1990): Partially Shared Views: A Scheme for Communicating among Groups that Use Different Type Hierarchies. *ACM Transactions on Office Information Systems*, vol. 8, no. 1, pp. 1-26.

Mariani, John and Wolfgang Prinz (1993): From Multi-User to Shared Object Systems: Awareness about Co-Workers in Cooperation Support Object Databases. In H. Reichel (ed.): *Informatik - Wirtschaft - Gesellschaft*. Berlin Heidelberg: Springer, pp. 476-481.

Mullender, S. J. and A. S. Tanenbaum (1986): The design of a cpability-based distributed operating system. *The Computer Journal*, vol. 29, no. 4, pp. 289-299.

Ontos (1992): Ontos Users Manual. Ontos,

Prinz, Wolfgang (1992): Representing Authorization Information in the X.500 Directory. In G. Neufeld and B. Plattner (eds.): *Upper Layer Protocols, Architectures and Applications*. Amsterdam: North-Holland, pp. 301-317.

Prinz, Wolfgang (1995a): *Applications of an Organization Information Server*. COMIC Document GMD-1-16 .

Prinz, Wolfgang (1995b): *Architecture and Services of an Organization Information Server*. COMIC Document GMD-1-14 .

Prinz, Wolfgang (1995c): *Management and Implementation of an Organization Information Server*. COMIC Document GMD-1-15 .

Prinz, Wolfgang (1995d): *Organization Modelling with the X.500 Directory Service*. COMIC Document GMD-1-11 .

Prinz, Wolfgang (1995e): *Requirements for an Organizational Information System*. COMIC Document GMD-1-10 .

Rumbaugh, James (1987). *Conference on Object-Oriented Systems, Languages and Applications (OOPSLA)*, pp. 466-481.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson (1991): *Object-Oriented Modelling and Design*. Englewood Cliffs: Prentice Hall.

Stein, W. (1993): Objektorientierte Analysemethoden - ein Vergleich. *Informatik Spektrum*, vol. 16, no. 6, pp. 317-332.

Suchmann, Lucy (1993): Do Categories have Politics? In G. d. Michelis, K. Schmidt, and C. Simone (eds.): *Proceedings of the Third European Conference on Computer Supported Cooperative Work - ECSCW '93*. Dordrecht: Kluwer.

X.500 (1993): *The Directory. CCITT Rec. X.500-X.521 | ISO/IEC Standard 9594.*

X.520 (1993): Selected Attribute Types, ISO 9594.Part 6 *The Directory.*

X.521 (1993): Selected Object Classes, ISO 9594.Part 7 *The Directory.*

Chapter 7:
An Organization Object Model

Wolfgang Prinz
GMD, Germany

The Modelling Domains

Organizational information includes a broad range of different information types. To identify the basic elements and structuring criteria for an organization object schema, it is sensible to separate the area of organizational information into different domains, where each domain highlights different aspects of an organization. Each domain is then represented in the OIS by a selection of characteristic organizational and relationship object classes. Figure 1 illustrates the separation applied for TOSCA, the motivation for which is explained afterwards.

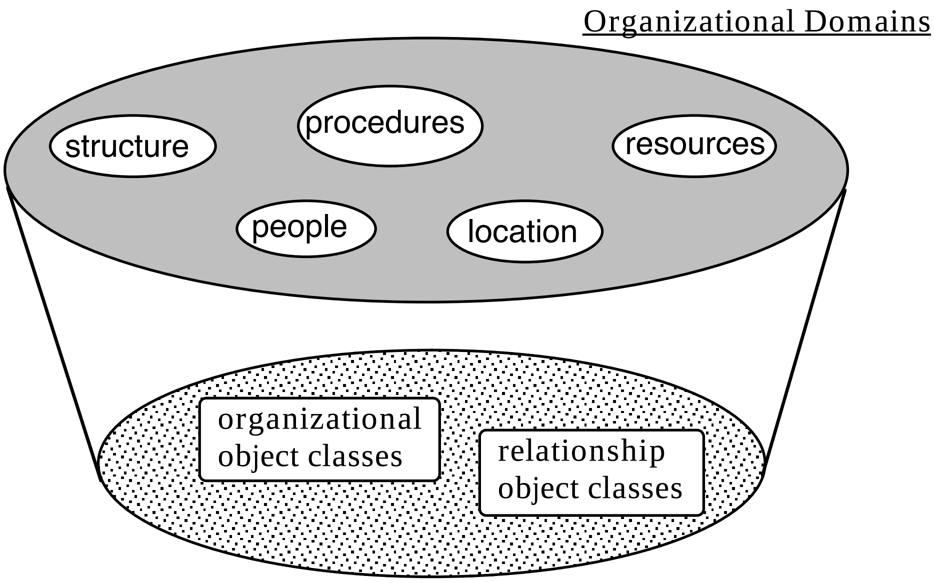

Figure 1. The modelling domains of the TOSCA organization model

The *structure domain* contains organizational entities and relationships which determine the structure of the organization. These can be regarded as the organizational skeleton into which the elements of other domains hook, in particular from the procedure domain. The *procedure domain* covers the organizational procedures and their actors. It specialises abstract descriptions of organizational functions in the form of structural or procedural roles which are themselves integrated into the organizational relationship-contexts.

The modelling of a particular organization requires the mapping of abstract positions, i.e. job or role descriptions to the members of the organization, i.e. people. These are considered from the *people domain*. This domain fills the organizational model with specific actors.

The *location domain* considers the geographical structure of an organization. These are the different locations of an enterprise or on a more detailed level its buildings or rooms.

The working material of the organization is envisaged from the *resource domain*. The resource viewpoint reflects on the objects that are processed within an enterprise or a group and the tools that are applied for the processing.

Obviously further domains than those presented here are possible, in particular if the social dimension of organizations are considered. However, those are not considered suitable for the modelling and representation by an OIS (Fuchs and Prinz, 1993).

The division of organizational information into these domains provides the basis for the definition of the organizational object schema for organizational objects and relationships. The following sections describe the resulting object schema and a detailed definition of the basic object classes is provided. The chapter concludes with an example section.

The Organizational Object Classes

This sections presents the object class hierarchy for the set of organization object classes of the OIS. In addition, examples of specific organization object class definitions will be presented. We start with a discussion of two examples for object class hierarchies for organizational modeling proposed by (Griggs, 1992) and (Hennessy et al., 1994).

Examples of Object Class Hierarchies

Griggs (Griggs, 1992) proposes an organization object class hierarchy which demonstrates two pitfalls that must be observed in the schema development process. The following figure illustrates the object class schema applied in his system for a visual representation of organizations.

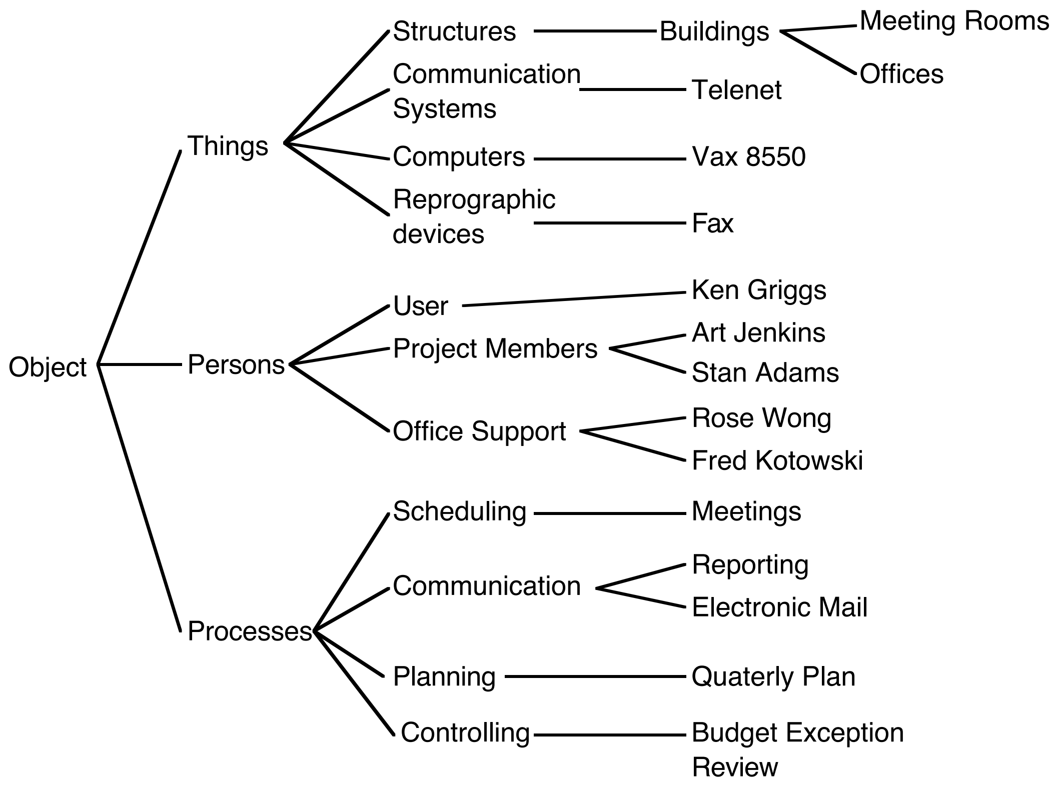

Figure 2. Part of the organizational object class hierarchy in (Griggs, 1992, pg. 217, fig. 7)

The objects *things*, *persons*, and *processes* build the top level of the class tree. The *thing* subtree contains objects that represent the physical entities of an organization, i.e. objects of the location and resource domain. The *Processes* subtree models objects of the Procedures domain. It is interesting to see that the object class schema does not reflect any elements from the structure domain, i.e. department, projects, etc.

Two peculiarities can be found in the *persons* subtree:

- Classes that model roles (*User, Project Members, Office Support* in this case) are modelled as subclasses of *person*. Thus, the modelling of abstract organizational functions is modelled as a refinement of a class that models individuals. From the organizational point of view this is not an "is a" relationship between two organizational objects, but an "is fulfilled" relationship.
- The association of people with roles, is modelled by representing individual people as subclasses of role object classes. Here, real-world objects which are expected to be represented as object instances are modelled as object classes. With this technique it is not possible to model persons who fulfil two different roles, since the object model does not support multiple inheritance.

The problem with such an organizational object class hierarchy design is the representation of organizational relationships as schema tree relations and the modelling of individuals as object classes. Although that might have been an appropriate modelling for the intended application, this schema cannot be used as a general approach for different organizational settings. Another important consequence is that the reorganization of the particular organizational setting would result in a reorganization of the entire object class schema. This is a poor approach to schema design. A model with appropriate relationships and the representation of actual people as object class instances is more adequate.

The Enterprise Information Service (EIS) (Hennessy et al., 1994) examined in (Prinz, 1995c) is based on the schema presented in figure 3.

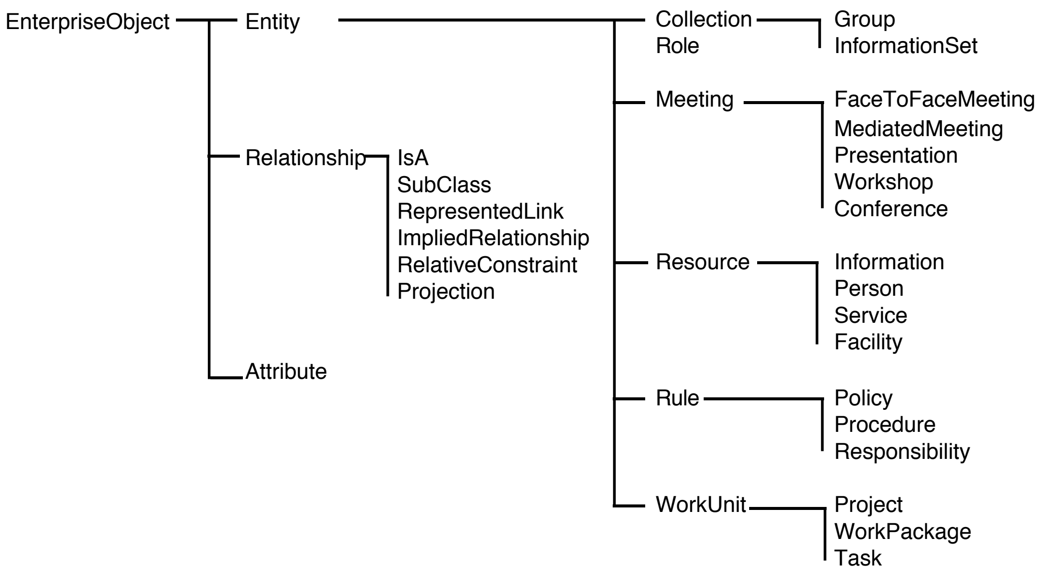

Figure 3. The EIS object class hierarchy (Hennessy, Harvey, et. al., 1994 pg. 134)

The EIS object class hierarchy distinguishes between three basic subtrees: *entity, relationship*, and *attribute*..

The *entity* branch models organizational entities. It is further refined into subtrees which cover the resource, procedure (*rule*) and structure (*work-unit*) domains. Two additional subtrees are defined for the modelling of *collection roles* and *meetings*. The *relationship* branch defines a set of relations for the modelling of relationships between objects of the *entity* classes. The attribute branch defines potential attributes of the *entity* classes. However, it is not further refined in fig. 5.3.

The following specialities of the OIS class hierarchy can be identified.

- The EIS class set does not contain a class for the modelling of organizational roles which are a common approach for the modelling of organizational functions. The class *collection role* is misleading in this context since it introduces a subtree for the modelling of sets of people (*group*) and information (*information set*). TO CHECK: any motivation for that fact??

- Persons are modelled as resources. This reflects a functional view on the role of people in an organization.
- The *WorkUnit* subtree is provided to model organizational structures. It contains classes for the modelling of a *project*, *workPackage*, and *task*. These are classes for the modelling of the dynamic structural components of an organization. Classes for the modelling of more static structures and organizational units, e.g. departments, are not considered.
- The detailed modelling of different meeting types by an own subtree of the class hierarchy is a speciality of the EIS object model. TO CHECK: what is it used for??
- The *relationship* branch contains a mixture of generic object model relations (*isA, subClass*) and application specific relations (*relativeConstraint*). Relationships which are typical for organizational modelling, e.g. role-occupation, sub-unit, or cooperation relations are not listed.

In summary, the EIS class hierarchy represents a well designed schema. The specialities listed above do not include any object-oriented modelling problems such as the previously presented Griggs model. They represent the EIS specific approach to organizational modelling.

The OIS Organization Object Classes Hierarchy

This section presents an organization object class model for the OIS which has been applied to model parts of GMD with TOSCA. However, major parts of the object class hierarchy include object classes which are general enough to be applicable for other organizations too. That has been demonstrated by its application for the modelling of parts of the German government for the POLIKOM demonstration (Pankoke-Babatz, 1994) as well as for other organizational settings.

The discussion of the examples in the previous section indicated that it must be distinguished between the object class relationship "is a" which is expressed both by the object class hierarchy and organizational relationships. Such relationships must not be expressed in the schema tree, as presented by Griggs, since this causes a redesign of the tree whenever the organization changes. Furthermore, instances cannot be represented in the class tree as Gibbs did with the modelling of individuals as subclasses of the *User* or *Project Members* classes. A more appropriate approach to the design of an object class hierarchy is presented by the EIS. The distinction between organizational *entities* and *relationships*. corresponds to the TOSCA approach. However, differences between the EIS and TOSCA model become apparent by a closer look at the TOSCA model which is presented in this and the following sections.

Further general recommendations for the identification of object classes can be found in (Rumbaugh et al., 1991, pgs. 153-156). The most applicable of these recommedations for organizational modelling is that to avoid redundant classes. It is tempting to model different organizational functions or concepts as distinct object classes instead of instances. The Griggs object model disucced in the previous section violates this recommandation by the modelling of particular people as subclasses of the user or project member classes. A further example is the modelling of different organizational roles by distinct subclasses of the role class, which is discussed below.

Based on these considerations the following organization object class hierarchy is proposed:

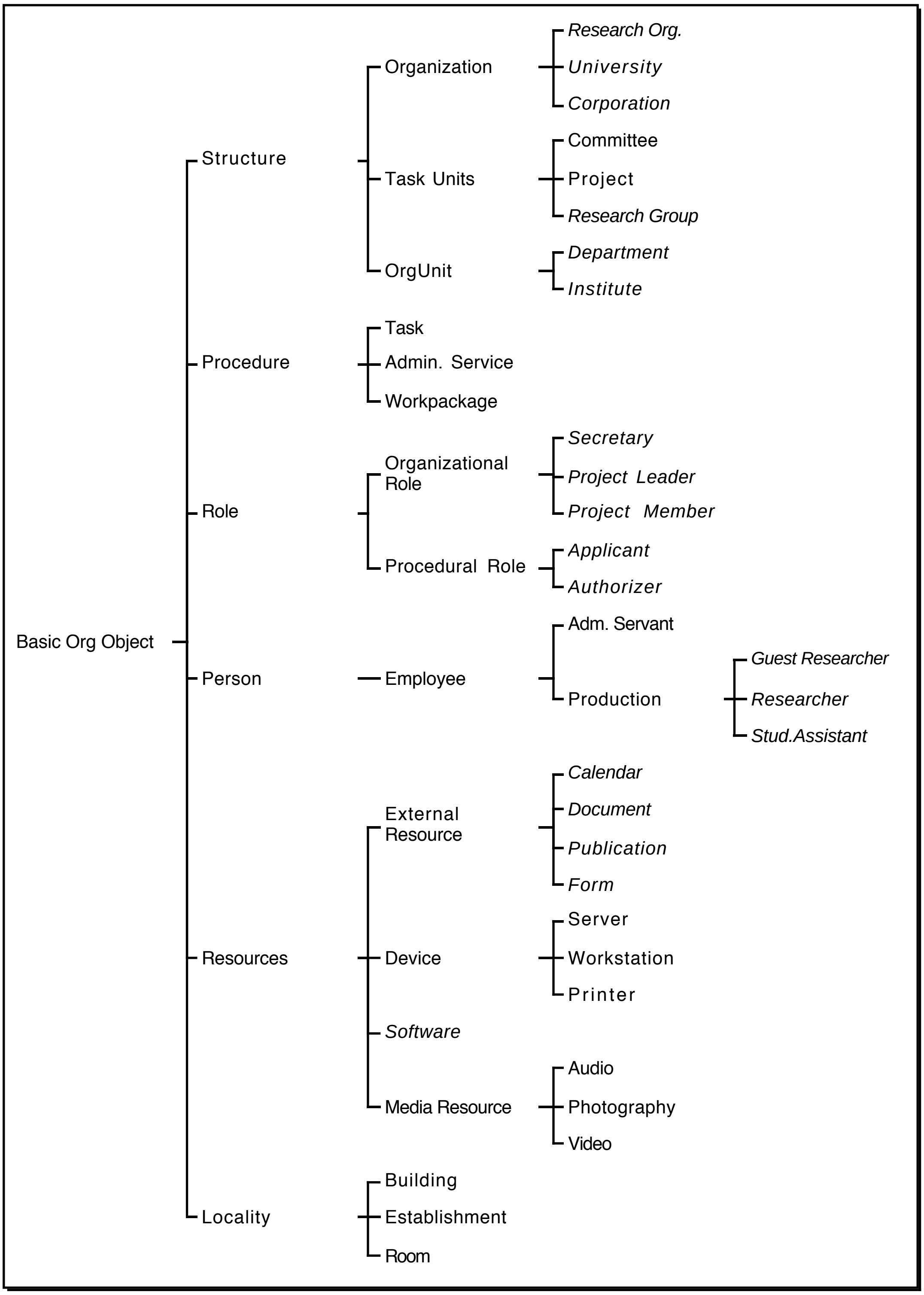

Figure 4. Class hierarchy of organizational objects

The tree diagram shows basic classes and organization specific extensions. Basic classes constitute the basic organization object model, e.g. *OrgUnit*, *Device*, or *Role*. Organization specific extensions to the schema introduce classes which are specific for a particular organization, e.g. the classes *Guest Researcher* or *Stud. Assistant*, which are specific to research organizations. The figure distinguishes them by different type styles[25]. This notation is also used for the schema trees that will be presented in the following sections.

The root of the class tree is the *basic organization object* class. This class provides the basic services required by all organization object classes for the management of organization objects. They are described in the following section.

The top level of the schema tree distinguishes between six different object classes. Each class is regarded as the root class of a subtree which contains object classes that model specific domains of an organization. These domains correspond to the domains presented in the first section of this chapter, with the exception of the *role* class. Although this class belongs to the procedure domain, it cannot be modelled as a subclass of the *procedure* class. All root classes are abstract classes, i.e. classes which should not be instantiated. Since they are designed as root classes their corresponding instances would not by expressive enough. Therefore only subclasses which contain more semantics of the object should be used for the creation of instances. This also supports a more selective use of the object query language (Prinz, 1995b) which allows the query for objects by their classes.

The structure subtree contains classes which can be applied to model the organizational structure of an enterprise[26]. An instance of the *Organization* class is used to represent basic properties of an enterprise, such as its name or a brief description. Normally just one object of that class exists for an enterprise model. This objects represents the root object for the modelling of the organizational structure. Organization specific subclasses of the Organization class are a *University* or *Corporation* class, which define additional properties specific for that organizational form. For the modelling of subunits of an organization two types of units are distinguished. Organizational units (*OrgUnit*) are applied for the basic structuring of an organization, such as *Departments* or *Institutes*. Task or goal oriented units which group employees together who fulfil a common task such as a *Project* are modelled in the *Task Unit* subtree.

The *Procedure* subtree contains classes that model steps of an organizational procedure. Obviously a variety of different classes are thinkable here, depending on the underlying procedure model. The DOMINO office procedure model (Kreifelts et al., 1984) would require classes such as actions, process documents, process channels, or action message. The Mac All model (Hennessy, 1990; Smith et al., 1991) would require classes such as person, role, information object, activity.

The specific subtree shown in figure 4 is based on the task model developed for the TaskManager (Kreifelts et al., 1993), since this cooperation support tool will be examined later as an application of the OIS in chapter 7. The basic unit of work in the task model is a task which is represented by the *Task* class. The further

---

25 Basic classes are typeset Helvetica and extensions are typeset *Helvetica-Italic*.

26 In the following the term "enterprise" is used to denote the real-world organization, while the term organization refers to the abstract organization model.

composition of tasks with people or roles and resources is expressed by appropriate relationships which are introduced in a following section. Two other classes listed in the *Procedure* subtree are *Admin.Service* and *Workpackage*. Objects of the *Admin.Service* class describe administrative services such as a technical infrastructure service, e.g. copy service or a software maintenance service, or an organizational service such as a cashier, standards archive, a bookkeeping service, or the travel agent. These services are normally associated with a role or an organizational unit. Objects of the *Workpackage* class can be applied to model different duties or procedural steps of a project.

The *Role* subtree branches into a subtree that defines organizational roles and procedural roles. The organizational role branch defines role classes in the context of the organizational functions. These are roles which fulfil a function in the context of the project or department, e.g. *secretary*, or a committee, e.g. *chair*. This understanding of organizational roles has originally been presented in (Prinz and Pennelli, 1992). It corresponds to the notion of structural roles in the ORDIT model (Blyth et al., 1993). The procedural role subtree contains classes which describe roles in a procedural context, e.g. an *application* for an organizational service such as *workshop organization*. Roles that appear in this context are roles such as *applicant*, *authoriser*, and *service provider*. These roles exist only in the context of the procedure description, thus they are called procedural roles. The lifetime of the binding of a role player to this role depends on the lifetime of the procedure, e.g. the binding of a person to a *reviewer* role in a reviewing process. Procedural roles can be occupied either by persons, or organizational roles. The notion of procedural roles corresponds to the term communicational role presented in (Prinz and Pennelli, 1992) and functional role of the ORDIT model**.**

In figure 4 both role-subtrees (*organizational* and *procedural role*) contain further refinements, e.g. *Secretary* or *Authorizer*. It illustrates the choice of modelling different roles as distinct object classes. To avoid an unnecessary proliferation of object classes, this modelling decision should only be made when each class models new functionalities. New classes should not be defined to differentiate between roles type only. In this case different roles should be represented by object instances of the *organizational or procedural role*

The *Person* subtree defines classes for the representation of the members of an organization, i.e. its *Employees*. This class can be subclassed into administrative or production oriented employees to describe the different jobs an organization offers.

The *Resource* subtree includes an *External Resource* subtree which contains classes that refer to external data. These classes model adapters which encapsulate external data or applications. Objects of these classes contain information where this data is stored or how these applications can be launched.

Finally the *Locality* subtree contains classes that model the geographical and spatial arrangement of an organization.

The next section will now present the services provided by the *basic organization object* class which can be extended by the different subclasses. Particular examples for subclasses are presented in the following sections.

Services Provided by the Basic Organization Object Class

The basic organization object class provides all methods needed for the administration and access of organization objects. This distinction of the methods by their purpose will be further used for a structuring of the method description. The event distribution model presented in chapter 5 requires the creation of appropriate event objects by all methods which result in a modification to the object base. Therefore the description of each method indicates if and which event object is raised by its execution.

Administration methods:	***Description***
`delete()`	*delete organization object* ***<this method raises a delete object event>***
`getLastModifiedDate()`	*get data of last modification*
`setAccessRight(newAccessRights)`	*sets the new access right on object level, overwrites the access rights defined on object schema level.*

One might wonder that there are no methods which allow the modification of attributes. Such methods are not provided by the object that encloses an object, but by the attribute object itself. The access method getAttribute(attributeName) returns an attribute object which provides all methods for the administration of the attribute.

Access methods:	***Description***
`getDistinguishedName()`	*get the distinguished name of the object, e.g. (DE, GMD, FIT, Kolvenbach Sabine)*
`getRelativeName()`	*get the relative name, e.g. Kolvenbach Sabine*
`getUserFriendlyName()`	*evaluates the user friendly name specification and returns the result, e.g. Sabine Kolbenbach*
`getAttribute(attributeName)`	*returns an Attribute Object*
`getRelationIdentifiers()`	*returns a list of all relation identifiers that relate the object to other objects*
`getRelation(relationIdentifier)`	*returns the relation object that is associated by the relation* `relationIdentifier`

`getRelObj(relationIdentifier)`	*returns all organization object that are associated by the relation* `relationIdentifier`
`getClassName()`	*returns the class name of the object*
`getGraph()`	*evaluates the graph specification and returns graph layout data that can be displayed by the browser*
`getAccessRightSpecs()`	*return the access rights defined on object level*
`getResponsibleAdmin(attr)`	*return the responsible admin defined ob object level for the given attribute*
`getAccessRightSpecs(newRightSpec)`	*set new access right*
`setRespAdmin(attr,newAdminSpec)`	*set new responsible administrator for the given attribute*

All methods check the appropriate access control rights[27] before they return a result.

These methods are sufficient for the basic processing of organization objects. However if additional functionality is required, the subclasses can be extended with new class specific methods, e.g. communication methods. Alternatively, special attribute type values (virtual and computable) can be used to add a specific functionality to organization objects, as illustrated in the next section.

Examples for Organization Object Classes

The definition of organization object classes is illustrated by the following examples.

The *Role* object class describes organization functions of an organization which are fulfilled by the employees.

```
class Role subclass of Basic Org Object

naming authority: Organization, OrgUnit

relative name: -- to be assigned at object instance level

constitutional attributes:   phone, fax, email,
                          competenceDescription

optional attributes:      description

constitutional relationships: occupy, role, service

optional relationships: task process, apply, substitute,
                     superiorSubordinate
```

27 Remark: the users identity is not provided for each method call. It can be internally requested from a session object (GMD-1-XX).

```
methods: getRoleOccupant
        getSubstitute
        [set|unset]SubstituteForward
        ...

modifySchema:   (distOrgName) -> TOSCAAdministrator

createSubclass: (distOrgName) -> TOSCAAdministrator |
               (distOrgName) -> subUnit -> TOSCAAdministrator

scope:    (distOrgName, *)

defaultAccessRights:

  create: (distOrgName) -> TOSCAAdministrator |
          (distOrgName) -> subUnit -> TOSCAAdministrator

  modify: (distOrgName) -> TOSCAAdministrator |
          (distOrgName) -> subUnit -> TOSCAAdministrator |
          this -> occupied_by

  read:   -- no specification: access granted to all

administrationResponsibility: this -> occupied_by

X.500objectClass: organizationalRole

userFriendlyName: relName+"("+ufn of this -> occupied_by+")"


description: "Instances of the role object class describe
             organizational functions and services which are
             provided, i.e.occupied by employees of an
             organization."

graphSpecification:
{
layout: Vertical

superiors:

[this];[this -> belongs_to];organizational unit where the role
                          belongs to

subordinates:

[this];[this -> occupied_by];employee who fulfills the role
[this];[this -> provides_service];service provided by the role
}
```

The following example describes the *Administrative Service* object classes.which is used to describe the services such as book-keeping, travel-assistance, project-administration, or workshop-organization provided by the administrative departments of an organization.

```
class Admin.Service subclass of Procedure

naming authority: Organization, OrgUnit
```

```
relative name: -- to be assigned at object instance level

constitutional  attributes:    phone, fax, email,
                          description, competence

optional  attributes:      service-hours

constitutional  relationships: service, belong

optional  relationships: task process, use

methods: getResponsibleRole
        getCompleteProcedure
        ...

modifySchema:   (distOrgName) -> TOSCAAdministrator

createSubclass: (distOrgName) -> TOSCAAdministrator |
              (distOrgName) -> subUnit -> TOSCAAdministrator

scope:    (distOrgName, *)

defaultAccessRights:

  create: (distOrgName) -> TOSCAAdministrator |
          (distOrgName) -> subUnit -> TOSCAAdministrator

  modify: (distOrgName) -> TOSCAAdministrator |
          (distOrgName) -> subUnit -> TOSCAAdministrator |
          this -> provided_by

  read:   -- no specification: access granted to all

administrationResponsibility: this -> provided_by

X.500objectClass: --

userFriendlyName: relName+"("+ufn of this -> belongs_to+")"


description: "The Administrative.Service object class models
             services provided by the administrative
             departments of an organization."

graphSpecification:
{
layout: Vertical

superiors:

[this];[this -> belongs_to];organizational unit where the role
                          belongs to

subordinates:

[this];[this -> occupied_by];employee who fulfills the role
[this];[this -> provides_service];service provided by the role
}
```

The Relationship Object Classes

The Relationship Class Hierarchy

Figure 5 displays the schema tree for the relationship class hierarchy that is proposed for the OIS. It is different from the relationships proposed for the EIS (fig. 3). The EIS defines relationship classes which describe abstract relationship types, e.g. *isA, subClass*. The TOSCA relationship classes define application-domain specific relations between organizational objects, e.g. *subUnit, substitute*.

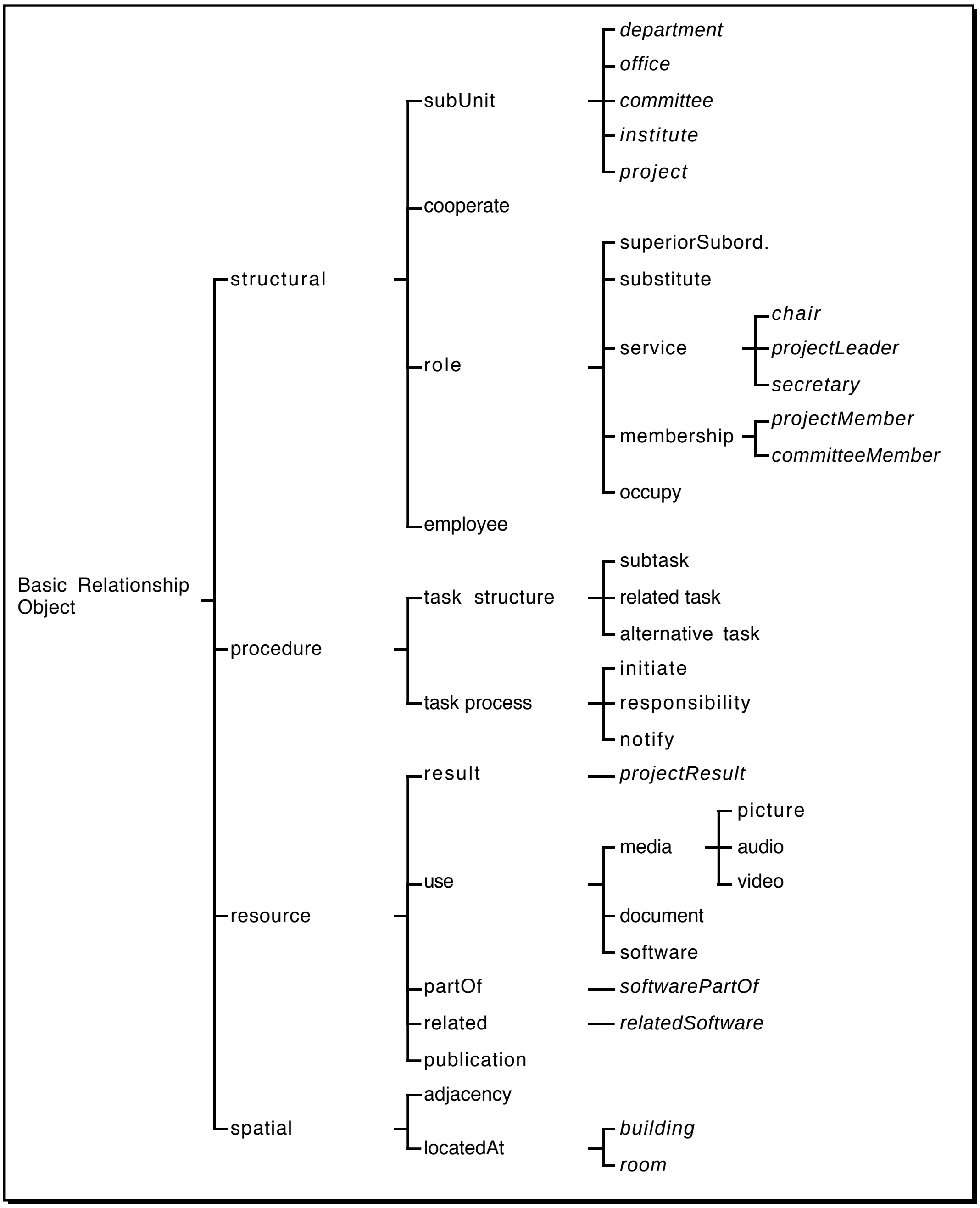

Figure 5. Class hierarchy of organizational relationship objects

The top level of the class tree distinguishes between structural, procedural, resource and spatial relationships. Relationships that relate to classes of the *Role* and *Person* subtree are included in the *structural* and *procedural* relationship subtree.

The *structural relationship* subtree contains a further subtree that defines classes for the representation of the organizational hierarchy. Objects of the *subUnit* relationship classes arrange organization objects of the *TaskUnit* or *OrgUnit* subtree into an organizational hierarchy according the organization chart of an enterprise.

Such relationships could be considered as aggregations (Rumbaugh et al., 1991, pg. 58), i.e. a special form of relationship between two classes. However, this distinction will not be further applied since it would impose a strict organizational hierarchy whenever aggregations are used which is not applicable for all organizations. Objects of the *cooperatio* class associate two organizational units which have no hierarchical, but a cooperative relationship.

Another subtree of the structural relationship tree is the *role relationship* subtree. It contains three distinguished relationships which are comparable to the structural relationships identified by the ORDIT model (Blyth et al., 1993): The *superiorSubordinate relationship* relates to a "power relationship". The *substitute relationship* expresses an partial equality between two roles and can be regarded as a "peer relationship". The *membership relation* associates a set of roles or people to an organizational unit, e.g. a committee or project. It can be regarded as a peer relationship too, since equality is expressed between the members. A "service relationship" is modelled by the *service relationship* which can be subclassed to some specific relationship such as *secretary*, or *chair*. The relations express, that the role provides a service to the organization object it is associated with, e.g. a secretary to a department, or a chair to a committee. The *occupy relationship* associates an employee to a role object. With objects of that class the framework of procedural and structural roles is filled with actual people, i.e. it "populates" the organization model.

The *employee* relationship associates an object of the *Employee* class or any of its subclasses with an organizational unit.

The *procedure subtree* branches into a subtree that describes structural aspects of a procedure and subtree that relates to procedural aspects. The structural branch relates to the task model introduced in the previous section 5.2.2. The *task structure* relationships allow the modelling of the aspect that procedures are decomposed into a list of related single tasks, which can be further sub-structured. In addition alternative task descriptions can be associated to each other. The procedural branch contains classes whose objects define who may initiate a task, who is responsible for its fulfilment and who should be notified about the task, i.e. who should be included as a member or observer of the task (this relationship will later be used for the provision of task templates to the TaskManager application - chapter 7).

The *resource* relationship subtree contains relationship classes for the instantiation of objects which associate objects of *Resource* class subtree with objects of any other organization object class subtree.

The *spatial* relationship subtree defines class that allow the description of the spatial arrangement of an organization.

Services of the Basic Relationship Class

The root of the class tree is the *basic relationship* object class. This object class provides the following methods for the administration of relationship objects.

Administration methods:	***Description***
`delete()`	*delete relationship object* ***<this method raises a delete event object>***
`getLastModifiedDate()`	*get data of last modification*
`setAccessRight(newAccessRights)`	*sets the `newAccessRight` on object level, overwrites the access rights defined on object schema level.*
`setAssocObj(relId, objSelector)`	*set the list of associated object for the relation identifier to those described by the `objSelector` expression*

The following table lists all access methods.

Access methods:	***Description***
`getAttribute(attributeName)`	*returns an Attribute Object identified by `attributeName`*
`getRelSpec(relationIdentifier)`	*returns the object selector expression for `relationIdentifier`*
`getRelObj(relationIdentifier)`	*returns all organization object that are associated by the relation `relationIdentifier`*
`getClassName()`	*returns the class name of the object*
`getAccessRightSpecs (relationIdentifier)`	*return the access rights defined on object level for `relationIdentifier`*
`getResponsibleAdmin()`	*returns the name of the responsible administrator for that relationship (see § 4.5.2 for a description)*

All methods check the appropriate access control rights[28] before they return a result.

Examples for Organizational Relationship Classes

The following examples illustrate the definition of the Occupy and Subtask relationship classes.

```
relation class Occupy subclass of Role

src: {Role}, occupied_by

dst: {Employee}, occupies_role

srcValue: to be assigned at object instance level

dstValue: to be assigned at object instance level
```

28 The users identity is not provided for each method call. It can be internally requested from a session object (GMD-1-XX).

constitutional attributes: duration, time_constraint

optional attributes:

methods: isValid(time)

modifySchema: (*distOrgName*) -> TOSCAAdministrator

createSubclass: (*distOrgName*) -> TOSCAAdministrator |

defaultAccessRights:

  create: (*distOrgName*) -> TOSCAAdministrator

  modify: (*distOrgName*) -> TOSCAAdministrator |
  this -> occupied_by

  read: -- *no specification: access granted to all*

X.500attribeType: TO CHECK: roleOccupant ??

sorting criteria: surname

subsetConstraint: --

forwardedEventClasses: Create, Modify

threshold: 0.1

relation class Subtask **subclass of** Task Structure

src: {Task}, subtask

dst: {Task}, subtask_of

srcValue: *to be assigned at object instance level*

dstValue: *to be assigned at object instance level*

constitutional attributes:

optional attributes:

methods: isValid(time)

modifySchema: (*distOrgName*) -> orgUnit -> TOSCAAdministrator

createSubclass: (*distOrgName*) -> orgUnit ->
TOSCAAdministrator

defaultAccessRights:

  create: this -> subtask -> responsible_emp

```
  modify: this -> subtask -> responsible_emp |
          this -> subtask_of -> responsible_emp

  read:   this -> subtask -> responsible_emp |
          this -> subtask_of -> responsible_emp

X.500attribeType:  --

sorting  criteria:  taskDeadline

subsetConstraint:  --

forwardedEventClasses:    Create, Modify

threshold:    0.1
```

The Attribute Value Types

An attribute value type defines the domain of the attribute values and the operations that can be performed on the values. The following figure lists the schema of the attribute value types of the object model.

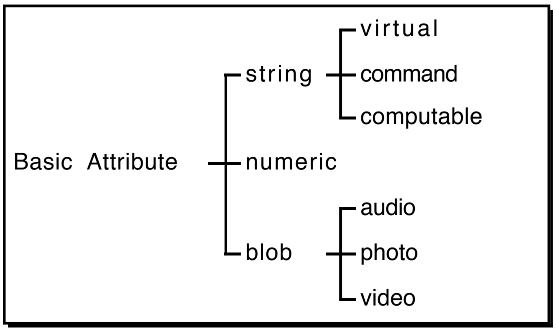

Figure 6. Attribute value type schema

The basic type defines all operations required for creating, accessing, and modifying attribute values and offers them via appropriate methods. Furthermore it provides the method `getResponsibleAdmin()` which returns the distinguished name of the administrator who is responsible for the administration of this attribute.

Administration methods:	***Description***
`addValue(newValue)`	*adds a `newValue` to the value list of the attribute* ***<this method raises a modify object event>***
`replaceValue(newValue, oldValue)`	*replace the `oldValue` of the attribute with the `newValue`* ***<this method raises a modify object event>***

`deleteValue(oldValue)`	*deletes the `oldValue` of the attribute* ***<this method raises a modify object event>***

A single attribute may contain more than one value. Therefore the modify methods must specify which old value shall be modified. If no oldValue is provided in the above methods, then all oldValues are replaced or deleted. All admistration methods must control the constraints defined for the attribute. That will often require the overriding of the administration methods by a subclass.

Access methods:	***Description***
`getValues(seperator)`	*returns a concatenation of all attribute values separated by the `seperator`*
`getValueIterator()`	*returns an iterator that allows the iteration over all attribute values*
`getResponsibleAdmin()`	*returns the name of the responsible administrator for that attribute*
`setResponsibleAdmin()`	*set new responsible administrator for this attribute*

The method `getResponsibleAdmin()` automatically returns the distinguished name of the responsible administrator. Administration responsibilities (AR) can be specified on different levels: the object class level, the attribute definition level, and the object level. On the object class level, the specified AR applies for all attributes of the class. This specification can be overwritten by the specification of an AR element in the definition of an attribute. If the object instance itself contains an AR specification for an attribute, this overwrites both. The `getResponsibleAdmin()` method evaluates the specification included in the object, attribute type or class definition according to the evaluation hierarchy. The basic relationship object class provides the same method. However, neither the relationship class, nor the relationship object provides administration responsibility information. Therefore this information is determined from the source object of the relationship. Since the source object of a relationship determines the organizational and administrative context of the relationship this appears reasonable.

In the following the specifics of the different attribute value types are described.

The value types of the *string* and *numeric* types are self explanatory.

The value of a *virtual* attribute type is used to store an expression of a simple language with the following functionality:

- concatenation of:
  - attribute values of the object itself or of an object which is selected by an expression of the object query language.
  - strings or characters as separators or delimiters between combined attribute values
- selection of alternative values in the case that an attribute contains no value.

When an attribute of the virtual type is accessed, the expression is evaluated and its result is returned. This functionality can be used for the following purposes:

- concatenation of several attribute values of an object to one, e.g. the composition of the content of the attributes, given name, surname, and title to an alternative name of an employee object. This technique is used to describe the user friendly name of an object class.
- retrieval of attribute values from other objects and their combination with the objects values, e.g. for the construction of an employees phone number by concatenation of the pre-code of the organizations number with the local number of the employee. This avoids redundant information in each employees entry.
- access to external information services such as the X.500 Directory. This is useful when information that is available in the Directory shall be partly integrated into OIS objects which describe a similar information object in the context of the organization model.

Example:

```
alternativePhoneNumber:

    phoneNumber of this->has secretary
    ","
    phoneNumber of this->has projectLeader
```

In this example a virtual attribute is used to define an *alternative phone numbers* attribute which returns the phone numbers of the secretary or the project leader concatenated with a colon.

This value type is called virtual to express the fact that its real value is transparent to a normal read access. It can be regarded as a special *derived attribute* (Rumbaugh et al., 1991, pg. 75)

The *command* value type contains a command that can be executed at the operating system level. This allows the access and execution of external services, e.g. document editors, communications programs. If the external service returns a value, this is returned as the attribute value. This allows a simple encapsulation of external services.

Command value type attributes may contain for examples calls to user interfaces of the world wide web. This allows a convenient access to other information services such as general information services or file servers which contain project related documents.

The *computable* value type contains an expression in an interpretable programming language. Therefore this type contains the additional method compute, which integrates an interpreter that executes the code stored as the attribute value. For example, the integration of the interpretable TCL (Ousterhout, 1993) language makes it possible to include simple programs which can interact with an external service. This provides means for a more sophisticated access to external services than the simple command value type.

Virtual, computable, and command value types provide additionally the methods `setExpression(expression)` and `getExpression()`. They provide access to the attributes expression without its evaluation.

The combination of virtual, computable, and command value types provides comprehensive means for the specification and extension of the behaviour of object classes and instances. The content of a virtual and computable attribute type can be defined on the class level and is then the default for all instances. Local modifications can then be made on instance level. Thus the behaviour of a particular object can be adopted to specific requirements without the need for a programming of new methods. However one should be aware that this contrasts to the philosophy of object orientation, i.e. that all objects of a class provide the same behaviour. Therefore this feature should be used with care only in those cases where minor adaptations are sufficient.

The *blob* (*b*inary *l*arge *ob*ject) attribute value type is used for the representation of various media types. The subclasses specify additional methods for the handling of the different media.

The Event Object Classes

Chapter 5 describes how events are used as a mechanism to provide awareness about changes in the organization. Thus the event object schema tree presented in figure 7 reflects the basic modification operations.

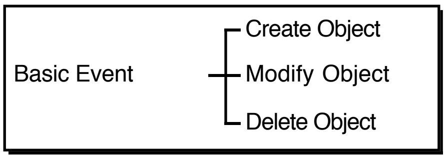

Figure 7. Schema tree of the event object classes

The basic event object class provides attributes to represent the information necessary for the event distribution algorithm and the following methods.

Methods	***Description***
`deleteEvent()`	*deletes the event object*
`copyEvent()`	*produces a copy of the event object*
`getIntensity()`	*returns the event intensity*
`reduceIntensity(resistance)`	*decrements the event object intensity by* `resistance`
`getAttribute(attributeName)`	*returns the attribute* `attributeName`*, e.g. originator, source object, time stamp*

These methods are sufficient for all events classes.

Chapter Summary

This chapter presented the organization object model of the organizational information server TOSCA. First, for each element of the organizational modelling framework an object class schema tree is proposed. Second, the essential administration and access methods for the base classes of each framework elements are defined. The following table demonstrates the contributions of the object model to the achievement of the requirements for an OIS.

Representation and provision of organizational structures, entities, and relationships	The organizational object model class hierarchy contains a certain subtree which reflects the structural components of an organization. The other subtrees contain objects class which model further organizational entities. A similar schematic structure has been applied to the organizational relationship class hierarchy.
Provision of information about organizational services and procedures	Both class hierarchies for organizational objects and relationship contain a particular branch which models organizational services and procedures. The application of these classes is demonstrated in chapter 8.
Communication support information	Appropriate communication attributes are defined for all concerned object classes. The application of these attributes is demonstrated in (Prinz, 1995a). The virtual attribute type supports the construction of alternative communication addresses or addresses from address parts.
Provision of awareness about organizational changes	The required event object classes which are needed for the application of event distribution model presented in chapter 5 are defined.
Data security and access control	All basic class definitions contain methods which allow the management and access to access control information.
Support for distributed management of the organizational information	Appropriate methods to set and get the responsible administrators of a particular object are defined for all basic classes.

Flexible and extendible organization model	The presented object schema provides a toolkit that can be adopted to different organizational settings. This is supported by a clear structure of the schema trees into the major domains of organizational information. as a prerequisites for a further development of the object model for specific needs and settings. The methods provide a basic set which can be extended for the purpose of specific classes. Thus, in case of a structural extension of the object model the basic methods are still comprehensive enough to support a reasonable application of the OIS.
Integration with the X.500 Directory	The virtual attribute type allows the retrieval of X.500 information. This can be then automatically integrated or added to the OIS information.
Integration of external applications and resources	This is supported by the command and computable attribute value type of the attribute class.
Architectural issues	The provision of the information about the responsible administrators by all basic classes separates the logical distributed administration from the technical distribution. Therefore the technical distribution can be administered independent from the organizational.

The aim of the proposed schema is the provision of a toolkit that can be adopted to different organizational settings. That has been achieved by a clear structure of the schema trees into the major domains of organizational information. This forms the prerequisites for a further development of the object model for specific needs and settings. The methods defined for the base classes provide the necessary functionality to administer and access instances of the organization object model. They provide a basic set which should be extended for the purpose of specific classes. Thus, if the object model is only structurally extended, i.e. the object model is extended only by classes which contain no additional methods, then the basic methods are still comprehensive enough to support a reasonable application of TOSCA. This has been examined in various demonstrations and at exhibitions were TOSCA has been adopted to different organizational settings.

References

Blyth, A. J. C., J. Chudge, J. E. Dobson, and M. R. Strens (1993). In S. Kapln (ed.): *Conference on Organizational Computing Systems, Milpitas, California*. ACM Press, pp. 216-227.

Fuchs, Ludwin and Wolfgang Prinz (1993): Aspects of Organizational Context in CSCW. In L. Bannon and K. Schmidt (eds.): *Issues of Supporting Organizational Context in CSCW Systems*COMIC - Deliverable Report 1.1, pp. 7-43.

Griggs, Kennth A. (1992): Visual Agents that Model Organizations. *Journal of Organizational Computing*, vol. 2, no. 2, pp. 203-224.

Hennessy, Pippa (1990): *Modelling Group Communication*. Computer Science, University of Nottingham.

Hennessy, Pippa, Paul Harvey, and Hugh Smith (1994): Support for Enterprise Modelling in CSCW. *Collaborative Computing*, vol. 1, no. 2, pp. 127-146.

Kreifelts, Thomas, Elke Hinrichs, and Gerd Woetzel (1993): Sharing ToDo List with a distributed Task Manager. In G. d. Michelis, C. Simone, and K. Schmidt (eds.): *Proceedings of the Third European Conference on Computer Supported Cooperative Work - ECSCW '93*. Dordrecht: Kluwer, pp. 31-46.

Kreifelts, Thomas, Ulrich Licht, Peter Seuffert, and Gerd Woetzel (1984): DOMINO: A system for the specification and automation of cooperative office processes. In B. Myhrhaug and D. R. Wilson (eds.): *Advances in Mocroprocessing and Microprogramming, Proc. EUROMICRO ´84 (Copenhagen, Aug. ´84)*. Amsterdam: North-Holland, pp. 33-41.

Ousterhout, John K. (1993): *An Introduction to TCL and TK*Addison Wesley.

Pankoke-Babatz, Uta (1994). *CSCW '94, Chapel-Hill, NC, USA*, pp. 10.

Prinz, Wolfgang (1995a): *Applications of an Organization Information Server*. COMIC Document GMD-1-16 .

Prinz, Wolfgang (1995b): *Architecture and Services of an Organization Information Server*. COMIC Document GMD-1-14 .

Prinz, Wolfgang (1995c): *Related Work in the field of Organization Information System*. COMIC Document GMD-1-17 .

Prinz, Wolfgang and Paola Pennelli (1992): Relevance of the X.500 Directory to CSCW Applications. In D. Marca and G. Bock (eds.): *Groupware: Software for computer supported cooperative work*IEEE Computer Society Press, pp. 209-225.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, and W. Lorenson (1991): *Object-Oriented Modelling and Design*. Englewood Cliffs: Prentice Hall.

Smith, H. T., P. A. Hennessy, and G. A. Lunt (1991): An Object-Oriented Framework for Modelling Organisational Communication. In J. Bowers and S. Benford (eds.): *Studies in Computer Supported Cooperative Work*. Amsterdam: North-Holland, pp. 145-158.

Chapter 8: Organizational Information and Coordination Support Systems

Wolfgang Prinz
GMD, Germany

Introduction

This chapter describes the interaction between a task management system and the server. The task management system can be regarded as a representative CSCW application, because it supports unstructured and structured cooperative work by the provision of mechanisms which allow its users to decide on the amount of control they want to enforce in the cooperative work. The flexibility of the task management system makes it furthermore applicable for the coordinated distributed administration of the organizational information server. The appropriate concept for the synergetic interaction between both applications is described in the final section.

Support for a Task Management System

Support of coordination support systems is a central aim of the organizational information service presented in this thesis. This section presents a concept for the support of a particular cooperation support system, the Task Manager (Kreifelts et al., 1993). In the following the Task Manager is briefly introduced. Then the interaction between the Task Manager and Tosca is described.

The *Task Manager* is based on a task-oriented cooperation model. The focus of interest are the tasks to be carried out and shared by participants in a cooperative enterprise. Aspects of activity coordination that are addressed by this model include specification of tasks to be carried out, responsibility for task performance, time constraints on task execution, resources needed for task performance, and dependencies that exist between tasks (Hennessy et al., 1993).

Flexibility and tailorability are achieved by integrating support for the planning and execution phase of task management and by allowing for adaptation to changes in the work place; this includes replanning, incomplete specifications, gradual refinement of abstract plans, negotiation and delegation of tasks.

Below the user interface layer, there is the activity management layer with the activity monitor and store which maintain a consistent view of the shared tasks for all users involved, local or remote. In order to achieve this, activity monitors of different user domains communicate with each other using X.400 message handling facilities of the message handling layer.

From this brief description it becomes evident that the integration of the Task Manager with an organization information service will result in a comprehensive user support. An integration concept is presented in the following.

Tasks descriptions are modelled in the OIS as so called task templates. A task template represents an abstract description of how a task can be carried out cooperatively in the organization. The term abstract indicates two aspects: first it means that the task template does not refer to any employees of the organization, but to organizational roles only. Second it indicates that the binding of a role to a task is expressed by a user specific expression. For example the responsible role that coordinates a room reservation is described as "user -> secretary", which is resolved to the secretary of the actual user when the task description is retrieved. Accordingly, the presentation of a task template at a user interface or the import into the task manager requires an interpretation of the task template beforehand.

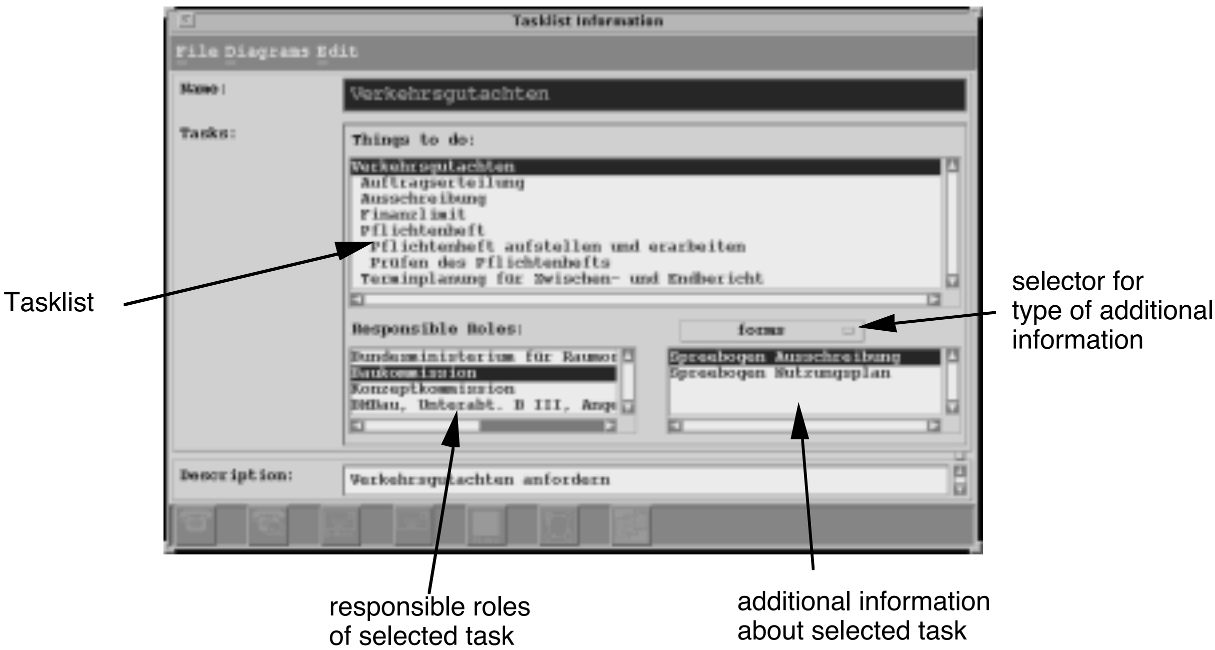

Figure 1. Presentation of a task template in the organization browser.

Figure 1 shows the presentation of a task template in the organization browser. The main list presents the subtasks involved in the task in an outliner format. The left sublist shows for a selected task the responsible roles. The right sublist shows the list of documents or forms involved in this particular task. If additional information exists for this task, e.g. an alternative task description, this is indicated and can be selected from a popUp menu. Any of the list items can be opened to show a detailed presentation of a role or form description for example. Since this way of task presentation provides an overview of the complete task, the visibility of organizational procedures is increased as demanded in (Schmidt, 1991).

The white board functionality of the organization browser can be used to comment on work-arounds or experiences one has made in carrying out a task. Using the multi-media capabilities presented above it is also possible to combine a task description with a video, either as a resource of the task or as an explanation of the task. This opens new ways of representing, offering, and explaining organizational guidelines to the organizational members.

The content of this window is collected by evaluating all relationships that associate the task object with roles, forms, etc. Thus this presentation shows already a user customised version of the task, i.e. different users will see different personalised task presentations, depending on their different organizational settings.

Although the presentation of this information is already very useful as a resource to initiate or plan a cooperative task, it becomes even more useful when it additionally can be transferred into an application that supports its further coordination. This is convenient in particular for routine task descriptions. With such functionality the OIS becomes a resource repository for a cooperation support system.

Two alternatives are possible for the import of task templates into a task management system. In the first alternative, the organization browser exports the task template to the task manager using an exchange file format. This means that the browser application stores the task description in a file from which it can later be imported into the task manager. In the second alternative only the name of the task template is exchanged between the organization browser and the task manager, by cut&paste for example. Then the task manager uses this name to retrieve the complete task template directly from the organization information server.

The second alternative has the advantages that no additional exchange format is needed between the organization browser and the task manager. However, it means that the task manager has to interface with the organization information browser. The fact that this interface is useful for other purposes is illustrated in the following scenario.

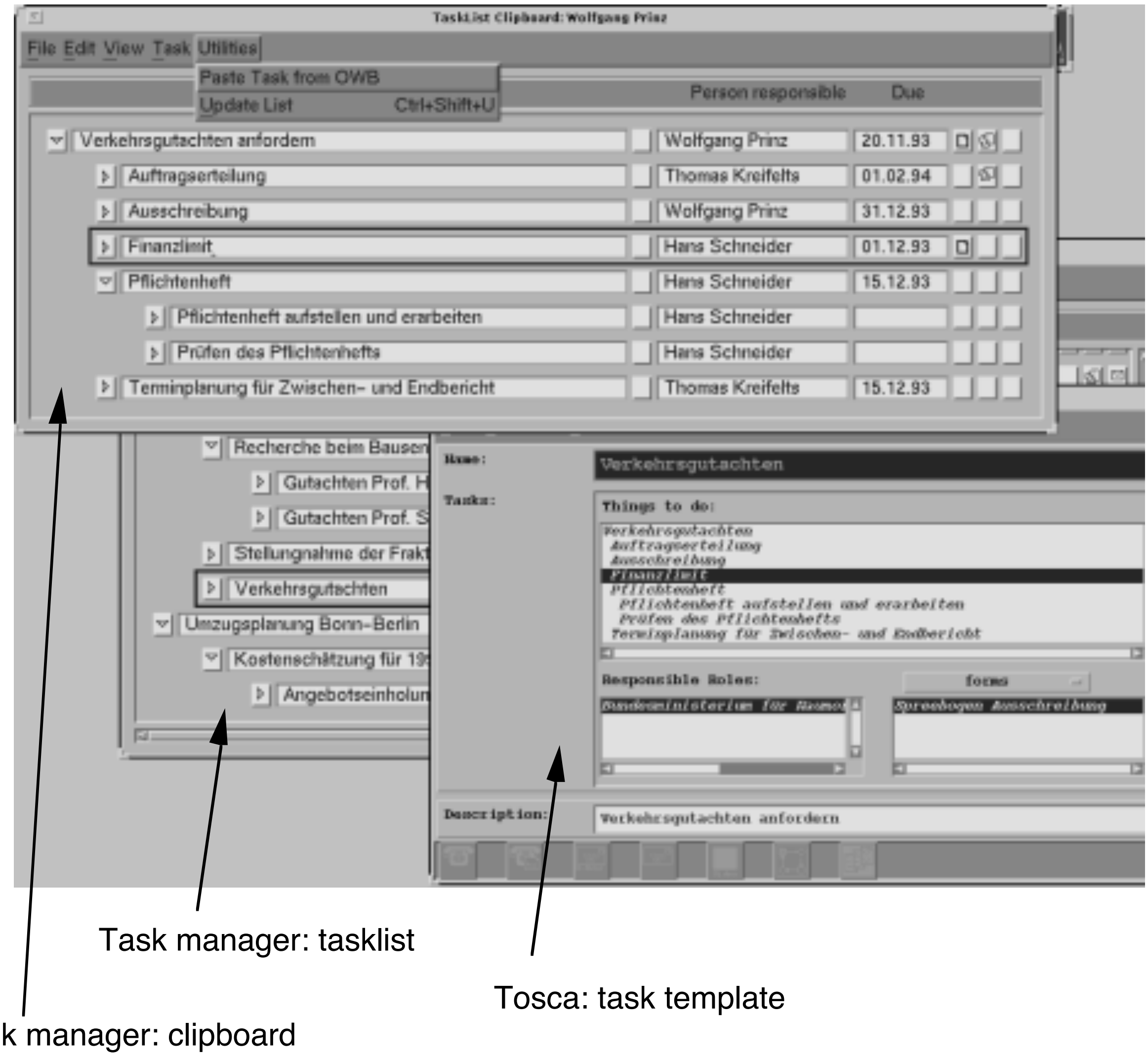

Figure 2. Inclusion of a task template into a users task list

After a user has found an appropriate task description in the TOSCA organization browser, the task name is pasted into a task list. This task name is used to identify the selected task template and the name of the task manager user is passed to the organization information server in order to retrieve a personalised task list. Then the complete task list including the predefined responsible role and resources (e.g. forms or documents) is retrieved from the organization server interface and included into the personal task list clipboard (fig.2). After the user has modified the imported task description to fit the actual needs, e.g. by deleting, inserting, or modifying responsibilities for involved subtask, the clipboard task list is included into the tasklist and distributed among the involved partners. In the further coordination process the task manager can access the OIS for address lookup, to resolve role descriptions, or to look for substitutes.

The task templates are understood as resources for the initiating users as well as the involved users to develop their own plan. They are not intended as a prescription of how a cooperative task must be carried out (Robinson and Bannon, 1991). Therefore other partners involved in the task execution may look for alternative task descriptions in the organization browser to substitute, extend, or modify their own part of the task process.

The benefits users get from an integration of a task management system with an organization information system can be summarised as follows:

- routine, or frequently used tasks description can be registered with the OIS and easily retrieved on demand
- inexperienced or new organization members find helpful task information in the browser and can utilise this for their own work.
- an imported task list already provides an assignment of responsible or supportive roles for tasks which involve administrative departments. Standard resources are also included where appropriate. Thus the user saves time normally needed for the collection of that information.
- alternative task descriptions can be searched and included when needed
- substitute information can be obtained from the OIS in the case that urgent tasks need to delegated to other cooperation partners.

But the integration is also of advantage for the developers and administrators of such a coordination support system:

- task list description can be centrally managed in the organization information server
- no additional address or resource directories need to be realised or managed
- the role to user mapping can be completely handled in the OIS

This section illustrated the advantages a coordination support system can draw from the integration with an organization information system. The following section considers the opposite case, i.e. the support of the coordinated management of the organizational information service.

Coordinated Administration Management

It is a special feature of the object model introduced in this work that it provides means for the specification of administration responsibilities. This section examines how that feature can be utilised to support a coordinated OIS administration management with the Task Manager.

Administration responsibilities (AR) can be specified on different levels: the object class level, the attribute definition level, and the object level. Applications do not need to know about this hierarchy in order to determine the name of the responsible administrator. The method `getResponsibleAdmin()` of the basic attribute type automatically returns the distinguished name of the responsible administrator.

This AR information can be applied to support two basic administration cases:

a) after the detection of a data inconsistency either by a user or by the organization information server the responsible administrator can be automatically informed about the problem.
b) when a new object is created, then different administrators will be responsible for the provision of correct data for that object. Then the AR specification can be used to inform all involved administrators about this new object and to request the needed data.

Each case is now considered in turn.

While users are browsing organizational information with the organization browser they may detect incorrect or missing information. For this case each window provides a menu that displays the list of all attributes defined in the object's class. This list is retrieved from the object's schema class. It contains also the class names of the relationships an organization object can be involved in. The user can now select the required attribute or relationship name from this menu list. Then the browser determines the responsible user calling the `getResponsibleAdmin()` method of the selected attribute or relationship object. The name and communication addresses are then displayed and the user may decide whether the administrator should be contacted directly, e.g. by phone or whether an administration request should be sent.

In the second case the user and the administrator will receive additional support from the OIS. First the user can provide an additional message to the administrator before the request is sent. When the administration request is confirmed by the user the following procedure starts. The browser calls the `adminRequest(...)` method of the data access object providing the following parameters: name of the concerned object, name of the attribute or relationship class, and the message provided by the user. The responsible administrator and the user is resolved internally by that method using the provided information. Then the preferred notify attribute of the administrator is read. If this indicates email as the preferred method then an administration request mail is composed and send to the administrator. This mail contains the needed information about the concerned object together with the message provided by the user. Since email can be considered as a standard infrastructure this solution is applicable in almost all organizations.

More sophisticated support can be provided if the administrator indicates the Task Manager as the preferred notification service. Then the data access object produces a task with the following content:

task name:	"Administration request for <object name>"
responsible person:	<name of the responsible administrator>
task member:	<name of the requesting user>
resource:	<call parameters for the OIS modify tool>
description:	<the message provided by the requesting user>

This task is forwarded to the activity monitor component of the Task Manager which distributes this task to the administrator as the person responsible for the task. Since the requesting user is listed as a member of that task it will be included in the users task. Therefore this user will be informed about the status of the task, i.e. when the task is done and the incorrect information is updated. If the administrator has further questions a task related email conference can be used for additional communications on the task. The resource coupled to that task contains the required call parameters for the OIS modify tool. This enables the Task Manager to call the modify tool in a mode that it displays the relevant object when the administrator opens the task resource. This is a very convenient method to provide the administrator immediately with the right tools to fulfil the task.

So far, the case that a user initiates an administration request has been considered. But the organization server might itself detect inconsistencies or problems during operation. This can occur in the following cases:

- the interpretation of a virtual, command, or computable attribute returns an error.
- the evaluation of a destination expression of relationship object returns an error or does not describe an existing organization object
- the evaluation of an access right specification returns an error.

In all these cases the same procedure as described previously can be initiated, with the difference that the user messages are automatically created and that no user is assigned as member to the task.

When a new object is created a similar support scenario can be provided. After the object has been created and initially stored, the data access object iterates over all attributes and mandatory relationships for which no data has been provided. Then all involved responsible administrators are notified and the missing information is requested either by email or by the initiation of a task. If tasks are produced then all administrators are included as members of this task. If an organization object is created that represents an organization member, i.e. an instance of a subclass of Person, then this member is also included as a member of the task. Thus, all of them get an overview of the status of the administration process.

The Task Manager application for the coordinated administration ensures that the requesting user and other involved administrators are informed about the status of the process. However, it is required that users who are affected by the modifications of the organizational information are also notified. In chapter 5 an event distribution model has been presented that provides the mechanisms to notify affected users about organizational changes. Thus the visibility of such an administration process is extended from the members of the process to the members of the organizational setting of the modified object. The following figure 3 illustrates this for a scenario where a user has requested a modification of a project object.

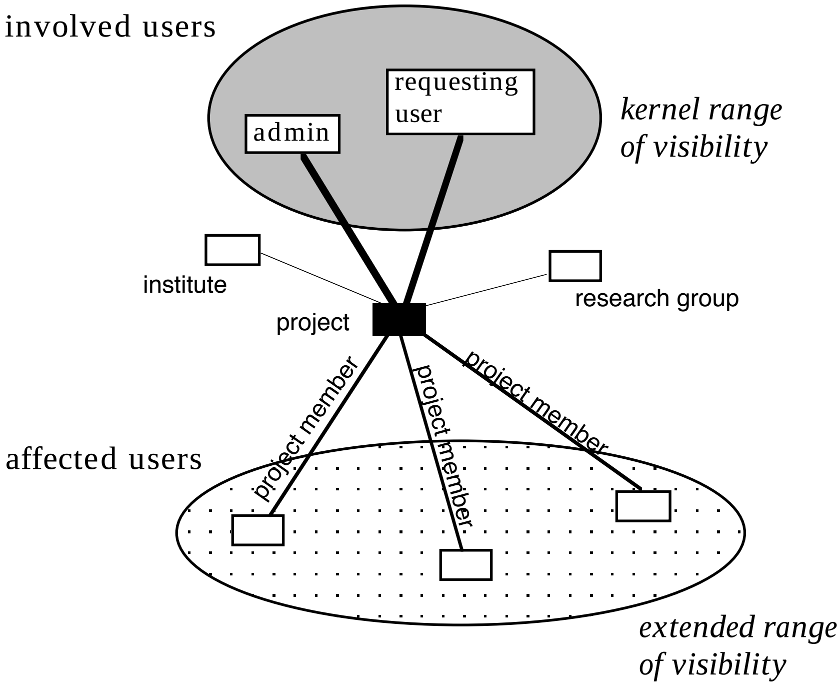

Figure 3. The extension of the visibility of organizational changes by event distribution.

The requesting user and the responsible administrator belong to the kernel range of visibility, since both are involved in the modification process. The application of the proposed event distribution model results in the generation and distribution of modify events. They are forwarded via the project membership relations to the members of that project, while other relations, e.g. to the institute or research group entries do not serve as a transport media (see the fowardedEventClasses property of the organizational relationship class in chapter 6). Then these people are notified according to their preferences specified in the "*notify*... " attributes. Thus the range of visibility of this modify operation is extended to these people.

This chapter presented the application of the AR specifications for a coordinated administration support. The proposed concept supports the initiation of administration request either by users or OIS services who detect inconsistencies. Furthermore the automatic notification of the responsible administrator is supported. An additional integration of a cooperation support tool, in this case the Task Manager, provides further advantages for both the initiating user and the administrator. Thus, this concept is an important contribution to the problem of a distributed coordinated support for the administration of large information bases, shown in the context of an organization information service. In addition the event distribution model provides appropriate means for the notification of users who may be affected by these changes.

Summary

This chapter presented three applications of the organizational information server. First, the integration benefits of the organization information service with a task management application for the purpose of a comprehensive cooperation support have been demonstrated. Second, a concept for the application of the task management to support a coordinated administration support for the OIS has been presented. All these applications demonstrate that an organizational information service plays an important role as a supportive service for users and applications in a cooperative setting.

References

Hennessy, Pippa, Thomas Kreifelts, and Ute Ehrlich (1993): Distributed work management: Activity coordination in the EuroCoOp project. *Computer Communications*, vol. 15, no. 8, pp. 477-488.

Kreifelts, Thomas, Elke Hinrichs, and Gerd Woetzel (1993): Sharing ToDo List with a distributed Task Manager. In G. d. Michelis, C. Simone, and K. Schmidt (eds.): *Proceedings of the Third European Conference on Computer Supported Cooperative Work - ECSCW '93*. Dordrecht: Kluwer, pp. 31-46.

Prinz, Wolfgang (1995a): *Applications of an Organization Information Server*. COMIC Document GMD-1-16 .

Prinz, Wolfgang (1995b): *Architecture and Services of an Organization Information Server*. COMIC Document GMD-1-14 .

Robinson, Mike and Liam Bannon (1991). In L. Bannon, M. Robinson, and K. Schmidt (eds.): *ECSCW ´91, Amsterdam, Netherlands*. Kluwer, pp. 219-234.

Schmidt, Kjeld (1991). In L. Bannon, M. Robinson, and K. Schmidt (eds.): *ECSCW ´91, Amsterdam, Netherlands*. Kluwer, pp. 1-16.

Chapter 9: The Milano System

Alessandra Agostini, Giorgio De Michelis, M. Antonietta Grasso
University of Milano, Italy

Introduction

The behaviour of contemporary organizations is increasingly characterized by their capability to satisfy customer needs. No standardized product or service can be successful if it fails to satisfy the customer requesting it in a particular moment, even if it respects high, prefixed, quality parameters. Therefore each business process an organization performs (in this context we consider business process and work process synonymous, in the former case emphasizing relations between performers and customers and in the latter relations among performers) represents a new story within which its members cooperate to satisfy the specific needs of a new customer.

Controlling costs remains a truly important objective, but paradoxically reduced costs must accompany higher customer satisfaction.

Standardized components and performances can be very useful from this point of view, but they must be embedded within business processes that are developed as services tailored to specific customers. Standards are useful in these situations as far as they are resources and not constraints for the creative cooperation of the performers.

Performers (newcomers as well as experienced professionals) therefore require help in concentrating their efforts to satisfy customer requests as they arise. They must on the one hand be liberated from all those activities not directly oriented to satisfying the customer (routine and administrative work), and on the other hand be assisted in performing the business process in which they are involved, as effective as possible increasing their *awareness* of the process itself, of its history and of the organizational structures within which they perform it (in short, of its organizational context) and providing them with any human and non-human resource they may need.

The relevance the previous statement attributes to the awareness of organizational context issue merits some comments, since it represents one of the crucial features of our approach to business processes.

Work is a situated activity (Suchman, 1987): its organizational context is characterized by two orthogonal dimensions: its history and its structure. On the one hand any work process is situated in its history: its already performed and ongoing activities characterize its uniqueness, its irreducibility to any predefined pattern. On the other, it is situated in a structure: the space its actors occupy, the common artefacts (tools, objects, procedures, ... (Robinson, 1993)) they share in it, the roles they play, the organizations to which they belong, etc. define the common space of possibilities where they perform it.

There is an interplay between the structure and the history of a work process: its structure is the outcome of its history, while its history is inscribed within the space of possibility defined by its structure.

Moreover, any new event modifies the context of a work process on both the historical and structural dimensions, while the future possibilities of a work process depend on both its history and its structure.

Finally, any work process is not isolated: its development interfoils with other work processes mutually influencing one another. In particular, their organizational contexts share, at least partially, the structural dimension as a way to let their actors share the experiences lived within them.

The bidimensional nature and the dynamism of the organizational context of a work process are the main causes of its complexity, of its failures, since they can induce its actors to perform in an ineffective way. In other words, overcoming the complexity of a work process greatly depends on the organizational context awareness of its performers. Coping with its complexity requires, in some cases, that they always know which performances are requested of them and how they can do them best (i.e., that the organizational context is transparent to their performances); in other cases, that they know what help they need to perform and where they can find it (i.e., that the organizational context is visible to them). Supporting the awareness of the organizational context means, therefore, supporting both its transparency and its visibility. Thus the organizational context awareness of its actors is the basic condition for effective performances within complex work processes.

The features of the work space shared by the performers (do they work in a common site or are they distributed in different distant sites; where are the tools they use, at their work places or not; which communication media are made available to them; where is the information on the process history, on the customer requests, on the organization to which they belong) play from this point of view a relevant role, since tool accessibility, richness of available communication media despite physical distances, and last but not least visibility and/or transparency of the organizational context greatly depend on work space organization.

In recent years computer-based work stations have dramatically changed the work space of most organizations, since on the one hand they have populated them with tools empowering their users with respect to communication, document management, information processing, etc., and on the other they have enriched it with a virtual extension through the creation of a multimedia space on their displays. Multimedia spaces can cancel out the physical distances between cooperating human beings and between them and the resources they need; they can reduce complex information processing activities to simple operations, liberating its users (more precisely, its occupants) from tedious clerical work; they can give plasticity to the work environment with respect to the ongoing activity.

It is our opinion that CSCW systems should be designed in such a way as to create multimedia virtual extensions of the physical work space in order to enhance the organizational context awareness of its users so that they can cooperate effectively within the business processes they are performing. Most of the prototypes and systems designed within the CSCW field have failed to reach this point: synchronous and asynchronous collaborative environments, synchronous and asynchronous multimedia communication handlers, workflow management systems, multimedia document and information sharing systems, have offered sophisticated and innovative tools supporting single aspects of cooperative work, but generally they are not able to support the awareness of their users with respect to the business process in which they are involved (Bowers, 1994), failing to improve the effectiveness of their cooperation.

We think that the multimedia virtual extension of the physical work space created by a CSCW system can overcome the limits mentioned above, if it is based on a model of the organizational context, so that it can offer the awareness support needed by its users. The role of models in designing CSCW systems is a controversial matter due to the contradiction that modelling work practices intrinsically hold. On the one hand, modelling is sought out and exploited as a way to embed the understanding that comes from the observation of a work practice in a support system so that it is reflected in the system behaviour, enriching the services it offers to its users; on the other, work situatedness determines the uniqueness of any work practice, its irreducibility to any predefined model. The apparent contradiction can be rephrased as follows: without a model the support system is poor, with a model it is rigid and normative. We think the bidimensional nature of the organizational context of a work process we have discussed above is able to offer a way to exit the contradiction: the historical context of any business process can be modelled by the time ordered records of its events, reflecting its unique development, while its structural context can be modelled in accordance with the rules (the artefacts) constituting the organizations to which its actors belong. The two dimensions of the model of a business process are mutually related: the events recorded in the historical model are characterized in terms of the structural model, while the elements of the structural model are situated in the historical record.

In this report we describe the Milano system, a prototype of a system supporting the cooperation within work processes. It is based on a bidimensional model of work processes, so that it can offer an awareness support in terms of both transparency and visibility. The prototype is currently under development at the Cooperation Technologies Laboratory of the University of Milano. Milano is an open system integrating a multimedia conversation handler, a workflow management system and an organizational handbook, as well as any tool its users may need to perform their activities.

The aim of the Milano system is twofold: on the one hand, it allows us to evaluate the approach to work processes we are developing at the University of Milano, so that its main features can be discussed at the behavioural level; on the other, it is a prototype showing the services that, in our opinion, new generation CSCW systems should offer.

After a short presentation of the work process model embedded in the Milano system (section 2), section 3 describes its components and the way they are mutually integrated. Section 4 presents the software architecture of the Milano system and shows how it allows to give an effective solution to the openness issue. A short presentation of the directions our future work (section 5) concludes the chapter.

The Model of the Organizational Context of Work Processes

A more detailed description of our approach to work processes can be found in another chapter of this deliverable (De Michelis, 1995). We can therefore dedicate this section to its aspects that are relevant with respect to the design of the Milano system.

Work process modelling has been approached by a variety of authors using a wide set of languages and formalisms (for a brief incomplete review of the work in this direction see (Malone et al., 1993)). Our approach shares its inspiration with the language/action perspective developed by Terry Winograd, Fernando Flores and their co-workers (Winograd and Flores, 1986; Medina-Mora et al., 1992), characterizing a work process on the basis of the relationships between the conversational practices of its participants and the activities they perform. But it reduces neither communication practices to any pre-defined conversation model, nor the relationship between communication and action to a relationship between one single two-persons conversation and one single activity.

Work processes embed any routine and new activity within the communication - the conversations, using Winograd and Flores (1986) terms - occurring between its performers and between the latter and their clients. There is no separation between routine and creative work: any successful work process intertwines them so that a procedure is started when needed within an ongoing conversation and a conversation is opened when needed from an ongoing procedure.

Work processes, therefore, are not characterized by the activities (the input-output transformations) performed within them but by the communicative practices, embedding them, between those who request a performance and those who perform it. A work process associates to its performers the commitments to action they have taken and eventually fulfilled and/or the conversations they have performed and they are performing.

Conversations are contexts of communication within which, when necessary, activities to be performed are defined as mutual commitments by participants. No matter if the activities to be performed are newly defined within the conversation or are instances of a predefined procedure, they are defined in a linguistic context. When a message is sent, when an activity is performed, they are situated respectively within one current conversation and within one current procedure, but they are also related to other ongoing or past conversations and procedures; in short their meaning depends on the history of the work process within which they occur. Moreover, the events of a work process are situated in the organizations of its respective participants, defining their roles, the resources available to them, the constraints they are subjected to in their behaviour; briefly their meaning is given in terms of the common artefacts characterizing their organizations. As we have claimed in section 1, the organizational context of a work process is bidimensional: it has both a historical and a structural component. The multiplicity of work processes in which any person is simultaneously involved does not allow us to assume a unique context for her work - any work process has its own context - while the dynamic generation of the organizational context of any of those work process does not allow us to assume a fixed context for it - the context of any work process is changing in time.

Multiplicity together with dynamism makes the organizational context of any work process (partially) opaque to its participants: none of them can be fully aware of what is going on, and therefore she requires information about it.

From the point of view of the relational definition of work process sketched above, the bidimensional nature of the organizational context is adequately reflected in the distinction between two components: the *inner organizational context*, whose changes are provoked by the history of the work process itself, and the *outer organizational context*, whose changes have their cause outside the work process, within other work processes (for a detailed explanation see (Agostini et al., 1995)).

The inner organizational context models its historical dimension as a partial order of the records of its past and expected events, while the outer organizational context models its structural dimension as a repository of the artefacts characterizing the related organizational structures. The events of the inner organizational context are represented making reference to the artefacts recorded in the outer organizational context, so that the participants of a work process may browse through the information characterizing its context without any continuity solution between its inner and outer components.

While the outer organizational context can be considered as the context offered to the participants of a work process in order to develop it, the inner organizational context is the context created by the participants of a work process developing it. While being aware of the outer organizational context means being an experienced member of an organization, being aware of the inner organizational context means being a committed member of a cooperative work process.

The Milano Components

An Overview of the Milano System

The Milano system is a modular environment, whose components, currently in the process of development, are the natural and logic decomposition of work processes from the point of view, we have shortly described in the previous section; in fact each module highlights a particular aspect of work processes, that is communication, action, and structure of the organization. Therefore it integrates a conversation handler (the Milano Conversation Handler - MCH), a workflow management system (the Milano Workflow Management System - MWMS), and an organizational handbook (the Milano Organizational Handbook - MOH) into a system that supports its users to situate themselves, at any moment, within the work processes in which they are active, with the highest degree of awareness. Each module addresses the aspects that are peculiar to each of these work process dimensions, at the same time the integration of the modules enables each of them to take advantage of the services of the others.

The integration between the three modules is effective to the extent that the user is supported by an interface that creates a multimedia virtual work space helping her to manage the information and services provided by each module in a smooth way, without being overloaded by the multiplicity and diversity of objects she can access.

On the one hand, the interface must, at any time, present that portion of the multimedia information and services that is relevant within the work process under consideration (its organizational context), putting the rest in the background; on the other hand, the user must, at any time, be able to browse and process the different objects presented in the interface on the basis of the sense of what she is doing, as if they were implemented in a fully transparent way. Moreover, the user must also be able to browse and process its multimedia virtual work space at the meta level where she is looking at her whole agenda, for example, to organize her working time.

Finally, a proper user interface must allow its user to move from one context to another in a smooth way depending both on her ongoing activities and on the occurrence of events in her cooperation environment.

The Milano system answer to the above problems is based on three principles: first, every event (both of the communication and action types) is hyper-recorded with bidirectional links to the record of the event under consideration at the moment of its occurrence, so that the historical dimension of the organizational context is represented with both its time order and its causal dependencies; second, every reference to persons, resources and artefacts is automatically linked to their characterization in the organizational handbook in terms of roles, organizational dependencies, rules, so that the structural dimension of the organizational context is always accessible to give meaning to the occurring events; third, in any moment, given the object under consideration, proximity values are associated to all the other objects so that the objects of the organizational context of a work process are selected and brought forth in the ongoing interface, and work processes can be ordered on the basis of logical or temporal vicinity. One of the key factors for doing this is the adoption of the rules proposed by Tom Rodden in (Rodden, 1994) in connection with the automatic linkage described above.

If the logical architecture of the user interface of the Milano system has a complete definition, its presentation is still issue of research. In fact, the richness and variety of objects characterizing the organizational context of a work process, together with the changes of the interface induced by the switches between work processes, require that the user is well supported in understanding what is happening in the system. If this result is not reached, the Milano system will be too complicated to use. The release of the Milano system currently under development is adopting a presentation style that is not yet matching our severe requirements, but we are already working together with Marco Susani and one of his students at Domus Academy on a new more sophisticated presentation style (Susani and Solaroli, 1995). We hope we can soon adopt it to the prototype under development.

In our system every single component is designed so that it can be considered a specific artefact to be used and referred to during the work. Each function a person exploits is meaningful in the context of the work process to which it belongs, and people performing a particular activity must concentrate on it taking care of its details and disregarding, as much as possible, the context in which it happens. Therefore in our system both dimensions are taken into account, on the one hand supporting the process dimension, and on the other offering artefacts that can be used per se in order to support each aspect of work, i.e. the artefacts populate the space environment in which people act in an orthogonal way with respect to the environment itself.

Since the natural unfolding of work occurs switching among communication, action and search of information useful to carry on the work the artefacts must support these three aspects.

Summarizing we can underline the following main aspects characterizing the overall system:

- each event occurring in one of the three modules is related to the *organizational context* of the work process to which it belongs; moreover each artefact provides a peculiar context to the events generated in it;

- each module provides a peculiar view of the work processes cutting off one of their dimensions;
- each module results *simplified* - with respect to a design in isolation - as some services become unnecessary, even if the features offered by the whole system go beyond the sum of those of the single ones;
- each module provides a switch to the others, maintaining the links among the occurred events;
- each module has been designed considering flexibility as a main objective, leading to the set of functionalities we describe in the next sections.

As we have claimed in section 1, the Milano system embodies a work process model based on the approach presented in section 2. In the following the basic elements characterizing the model underlying the Milano system - namely work process, conversation, activity template and procedure - and their mutual relations are presented in a synthetic way.

In our definition a work process is a relation involving some participants, sometimes requesting some services meeting specific conditions of satisfaction and sometimes performing them (De Michelis, 1995). On the basis of the services they request and perform, participants play within the work process the roles of customer and performer, respectively. Service requests and performances are plunged in a wider and less finalized communication (conversation network) creating the common background of the participants to the work process, thus allowing them to give sense to the service requests and performances. A work process is constituted by conversation(s) made upon by communicative events (of multimedia nature), embodying, when necessary, activity template(s) devoted to sharing precise conditions of satisfaction for the requested and performed services. Unlike in the Flores four-phases model (Medina-Mora et al., 1992), we propose a definition of conversation and negotiation (De Michelis and Grasso, 1994) taking into account the communicative context in which the formal definition of commitment is, if necessary, stated. Commitments to activities are negotiated and taken within conversations, but the relation between them, as pointed out in (Bowers and Churcher, 1988), is not a one-to-one relation, so reducing a conversation to a commitment can reduce the possibilities open to its participants. But making a commitment negotiation explicit by means of an activity template may be very useful, especially if we want a successful completion of it.

Let us now focus on what happens during the phase devoted to performing a requested service. On the one hand a procedure could already have been specified in the organization in order to support the kind of request defined by the customer. This is the case of activities like the ones performed in insurance companies or in the administrative offices of almost any firm. A library of procedures is normally available from which to choose the appropriate one. A procedure definition offers some benefits (for more details compare (Curtis et al., 1992)): the partial order of activities needed to perform a service is already defined, as are the responsibilities of its actors; the assignment of tasks to the actors does not need any negotiation, since they know their role and how to carry it out; the coordination of all actors is planned in a standard way. On the other hand, it could be the case that the requested service requires an ad hoc planning and therefore the activation of new negotiations in order to define the partial order of activities needed to perform the requested service, to assign each activity to a person, defining its specific conditions of satisfaction. In this case a work process generates a network of related activity templates, embedded in one or more conversations. Sometimes the plan can be defined at the beginning of the work process, sometimes the activities to be performed are identified only during the execution of the work process itself. Summarizing we can distinguish three main kinds of work processes:

- a *Routine* work process where the flow of activities to be performed is known in advance and it could be defined a priori like in bureaucratic procedures;
- a *Planned* work process where the flow of activities to be performed can be defined only at its beginning and changed, if necessary, during its execution;
- an *Evolutionary* work process where the flow of activities to be performed is created day by day.

We want to underline that real work processes generally appear as a *composition* of the above ones, i.e. procedures need an extra planning in order to respond effectively to a given request, while planned and evolutionary work processes sometimes could exploit procedural schema for some routine parts.

The Milano Organizational Handbook

Some work has been done to face the problem of representation of the organizational structures and entities (see for example WooRKS in (Ader et al., 1994), or for an exhaustive survey (Prinz, 1995)). Main objectives of this kind of system are the definition and the handling of knowledge about organizations and the resources which are relevant for the support of cooperation; actually they enhance the level of abstraction provided for example by repository systems (Jarke et al., 1992). For instance the TOSCA system (Prinz, 1993) provides both a predefined set of organizational objects and the possibility to refine them in order to meet the specific requirements of an organization.

These organizational handbooks can be used to handle all details about an organization in order to provide them to other CSCW applications. In fact usually they are flexible and general enough both to share the basic organizational objects with applications and to provide services they need. Moreover, even if they are not integrated with other CSCW applications, they still can provide the user with useful functionalities to browse and easily retrieve information. But generally they disregard the problem of introducing a model of cooperation units that allows us to guide the user in the information retrieval, selecting just the meaningful one. Therefore in the Milano Organizational Handbook (MOH) we can distinguish two classes of functionalities, the first one providing the knowledge of organizational structures, the second one the work process view of the organizational context. In other words the second part augments the static information with dynamic information from work processes. Thanks to the work process view, we can look at an organization through its work processes and therefore (starting from them) retrieve the meaningful organizational context. During the first design phase of the MOH (Agostini et al., 1995) the TOSCA system (Prinz, 1993) has been used to play the role of organizational handbook of the institutions involved within a work process, thus becoming the base for the organizational context model of work processes. In the following section we explain in more detail the MOH objects and relations among them that are defined to provide users with organizational information.

Objects and Relations among the Organizational Objects

From the logical splitting of the MOH in two parts we can distinguish two categories of objects (see figure 1, where the main objects are shown): the basic objects[29] devoted to modelling the structural information of the organizational context (such as resources, localities, roles and so on) and the peculiar objects devoted to the representation of the work process model of the organizational context (such as work processes, procedures, conversations, and so on).

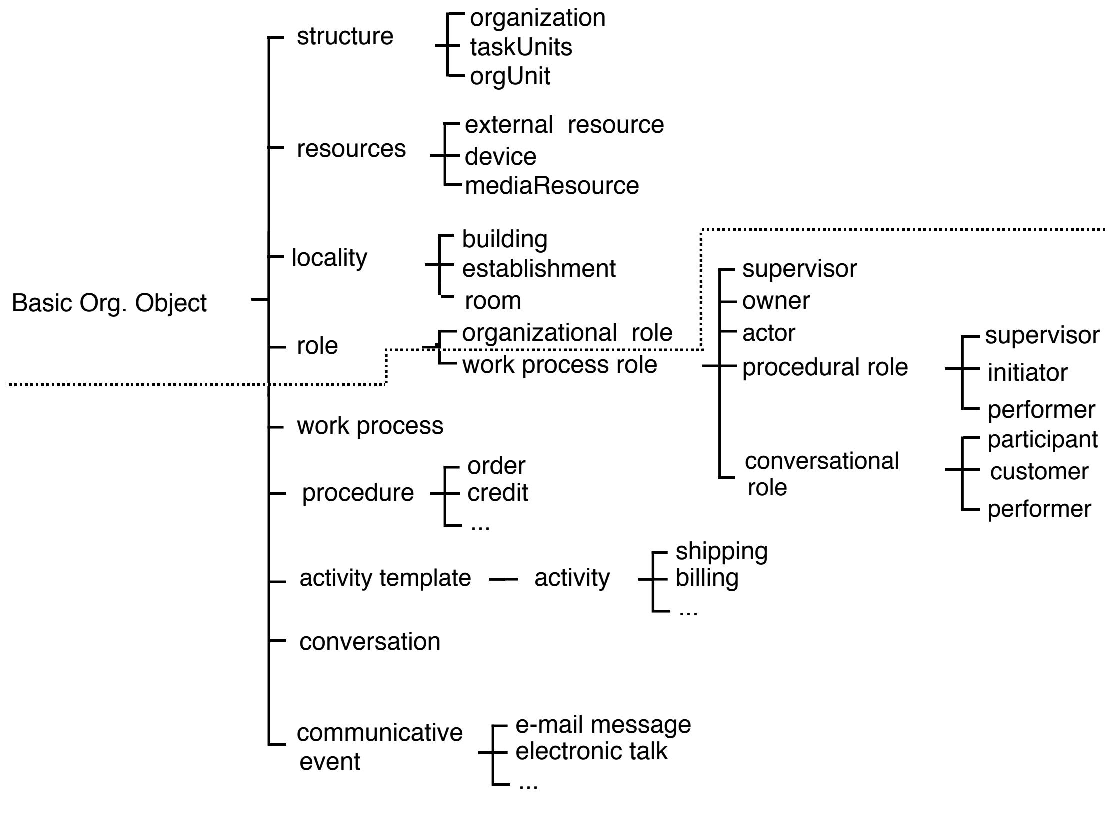

Figure 1. Class hierarchy of organizational objects.

29 The first category of organizational objects is a revised version of TOSCA objects (Prinz, 1993).

Let us remark that the definition and handling of work process objects such as conversations or procedures are delegated to the appropriate components (MCH and MWMS respectively in this case); in the sense that the MOH offers the storage and retrieval functionalities (for example the storage of conversations in the organizational handbook or the resolution of organizational roles) to the other components; whereas the MOH and the MWMS take care of when, where, how and according to which rules new objects are created, new events must be traced and so on.

The peculiar issue of this component is the representation of the dynamic information regarding the organizational context of work processes; this aspect is obtained through the definition and handling of relations among the objects characterizing the work processes. Therefore in the following we will describe in more detail the relations linking the work process objects allowing us: to navigate from an object to the object to which it is related; to situate any simple object within the more complex one where it has occurred; and more generally to cut off a work process view on the organizational handbook.

We distinguish between operational relations (figure 2) which characterize how any of the work process objects is created within a work process, and structural relations which characterize how they can be mutually embedded (figure 3). Operational relations are traced in the system and they allow us to create the structural ones, which provide paths of object reachability.

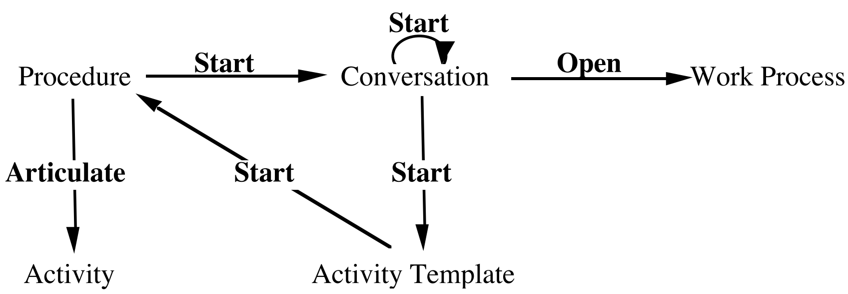

Figure 2. Dynamic operational relationships of work processes.

With regard to the operational relations, any conversation *opens* a work process (virtually, if the conversation starts from an already open work process, it is not necessary to open a new work process) since an activity template can start from it. Each conversation can start an arbitrary number of related activity templates, therefore belonging to the same work process. When a new conversation is *started* from an existing one, a new process opened within this conversation is indeed a sub-process. A new activity template can be *started* from a conversation every time we want to make explicit the responsibilities and the conditions of a commitment. Sometimes the path to follow in order to perform the service requested within an activity template is already defined as a procedure; in this case the activity template *starts* (the execution of) the previously defined procedure. When a procedure is started the activities in which the procedure is *articulated* - via its formal definition -

can be executed step by step. Finally, every time a breakdown occurs during the development of a procedure, the participants of a work process can *start* a new conversation in order to handle the exceptional situation and to overcome the deadlock.

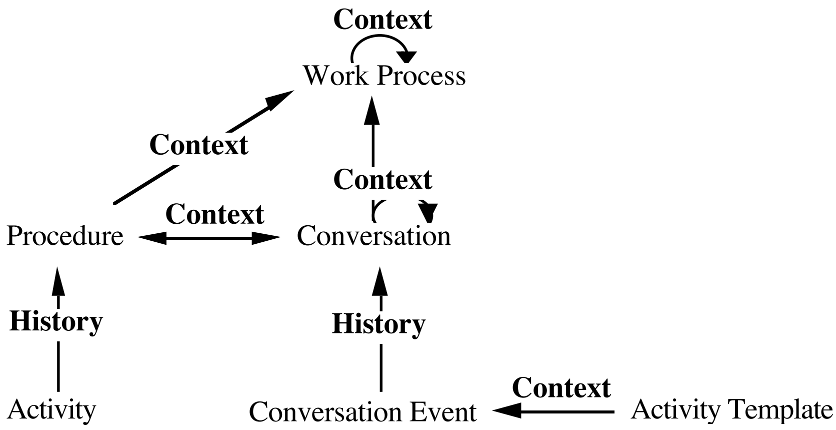

Figure 3. Dynamic structural relationships of work processes.

With respect to structural relations (figure 3) we distinguish between context and history relations. Through context relations it is possible to navigate from an object to the object to which it is related, whereas through history relations it is possible to situate any simple object within the more complex one where it has occurred. Contextual and historical relations therefore allow us to cut off a work process view on the organizational handbook as the basis of its organizational context. For example, when a procedure starts a conversation the latter is contained in the procedure context. Each relation contributes to the context of a work process, that is, the context of a work process may be enlarged scaling it as needed.

The Milano Conversation Handler

The language/action perspective has had a significant impact within the computer supported cooperative work field; nevertheless, both its theoretical foundations and The Coordinator™, have been subject to criticism in a passionate debate lasting almost ten years. A list of criticisms and positive issues has been analysed in (De Michelis and Grasso, 1994) and a first proposal of system implementation based on them has been done. The MCH presented here (De Michelis et al., 1995) is a further refinement towards a new exploitation of the language/action perspective, taking into account the list of issues analysed in (De Michelis and Grasso, 1994). In particular the MCH aims to contributing to the debate on the language/action perspective with the following points:

- the association of a speech act to an utterance is recognized not as a one-to-one mapping, but as the effect of an interpretation from a viewpoint. The system supports this assuming the communication as free as desired, but

offering the means to make explicit the viewpoint of a person requiring something to be done or offering to do something;
- the conversations to which the commitments are linked (by means of activity templates) support the process of negotiating the agreement of meaning;
- the situatedness of the activities is obtained in two ways: first, by linking them to the conversations, and secondly, by inserting the conversations in the work processes.

The conversation is the main unit of communication in the MCH, acting as a folder collecting communicative events from a group of participants, which can be enlarged upon request from the initiator of the conversation. The conversation is the common context the participants share in order to give meaning to the communication; a conversation holds the information about the state of each event, highlighting which events can be considered part of the context of communication and which not. That is, for each event of the kind of electronic messages it is possible to know if the recipients got the messages or not and which messages have been received when a new one is generated. Let us recall that a conversation is made of communicative events, meaning that not only electronic messages compose it, but a variety of other media can be used to produce a communication. An example of a system implementing multimedia capabilities is described in (Agostini et al., 1994a). Up to now electronic mail messages have been implemented, but we plan to extend the set of supported events in order to include other media such as electronic talk, face to face meeting, phone call, and so on. Each event is characterized by the possibility to attach enclosures of various types. Besides electronic documents such as spreadsheets and graphical pictures, the activity template is a special kind of enclosure that can be created each time there is the need to formalize the terms of something to be performed. It is generated within the context of a conversation and when it is accepted it becomes an object of the To-Do List of both the customer requesting it and the actor who is in charge of the activity.

Proposal & Agreement	They are carried on through various communicative media (email, chat, face-to-face talks, meetings...), as free exchange of information (text and objects). The creation of activity templates contextualized in the exchange is devoted to supporting the process of listening: an agreement can be formalized filling in fields as needed (description, deadline, resources, state, etc.)
Performance	It is supported by an updating of the template (state, results)
Satisfaction	It is expressed by transition to a final state (implicit agreement), and/or exchanging messages (explicit agreement)

Figure 4. Mapping the Action Workflow™ loop in the MCH.

It contains a description, a state, and a deadline that can be changed by directly accessing the object. The two roles with respect to the activity generate some rules which time by time allow the possible changes. Moreover after its creation an activity can also be modified directly accessing the object; but this opens the related conversation and starts a new message notifying the changes (see figure 6; for a comparison with the Action Workflow loop see figure 5). Finally let us recall that another interesting link with the MWMS supporting the execution of procedure is related to the possibility of handling jumps in the procedure, by creating appropriate activity templates. This is explained in the section about the MWMS.

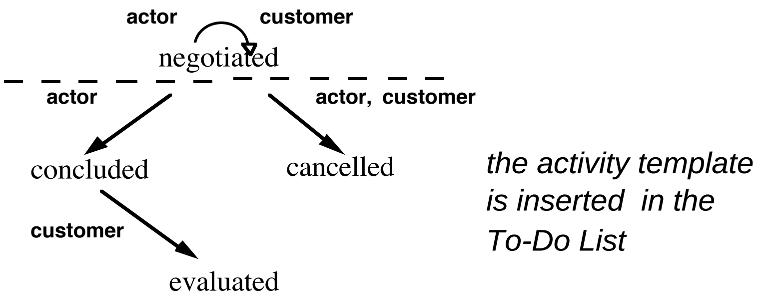

Figure 5. The states of the activity template.

The Milano Workflow Management System

Coupled to the conversation handler there is the Milano Workflow Management System (MWMS), facing the aspect of work complementary to communication, i.e. the action and its coordination. Main objectives of Workflow Management Systems are (Ellis et al., 1995):

> "to help organizations to specify, execute, monitor, and coordinate the flow of work items within a distributed office environment".

From this general definition we can distinguish in any WMS two main components:

- the *workflow model*, devoted to supporting the definition, analysis and simulation of the flow of work;
- the *workflow execution module*, devoted to supporting the execution of the flow of the work and the coordination of users in performing it.

For many years WMSs have been developed in the academic field (we want to remember here for example (Ellis et al., 1979; Ellis, 1982; Kreifelts et al., 1991)); moreover WMSs are currently offered (see for example (Swenson et al., 1994; Medina-Mora et al., 1992; Sarin et al., 1991)) and/or announced by many software and hardware suppliers. Despite their growing success, already existing WMSs exhibit severe limitations. Let us recall some of them. Exception handling mechanisms, a main issue for such systems, are generally either too restrictive with respect to the needs of users or too complicated with respect to their usability. WMS focus on routine procedures, disregarding ad hoc plans and evolutionary processes (see section 3.1). Workflow models are too complicated affecting in general the flexibility and the usability of the workflow execution module. This holds true in particular also for their static changeability (that is it is difficult to grant that the behaviours they permit are consistent with respect to a given specification); and their dynamic changeability (that is it is difficult to activate run-time changes in the procedure definition preserving the correctness with respect to a given specification of all ongoing instances).

We claim that the simpler the workflow model, the more flexible and adaptive the workflow execution module. That is, the two main components of a Workflow Management System are not merely inter-related: if the former allows simple models and complex behaviours, then the latter is able to support execution and changes in an effective manner. We have learned from our experience with the WooRKS-UTUCS (Agostini et al., 1994b) prototype that the coupling of a Workflow Management System with a well defined Communication Support System is a good starting point for providing the user with both a simple workflow model and an effective overall workflow execution module.

The workflow model of the MWMS is based on Petri Nets (Rozenberg and Thiagarajan, 1986), as many previous Workflow Management Systems. In particular it is based on a small subclass of Elementary Net Systems (EN Systems) whose elements are simple and highly readable models, and on the other allowing us - thanks to some nice mathematical properties of that class - to generate a large class of behaviours. For example the property described in (Ehrenfeucht and Rozenberg, 1990; Nielsen et al., 1992) allows us to switch freely from a Net System representation to a Transition System. An example of definition of an order procedure (the initial part) with this formalism is shown in figure 6; both the EN System and its corresponding Elementary Transition System (ET System) are drawn.

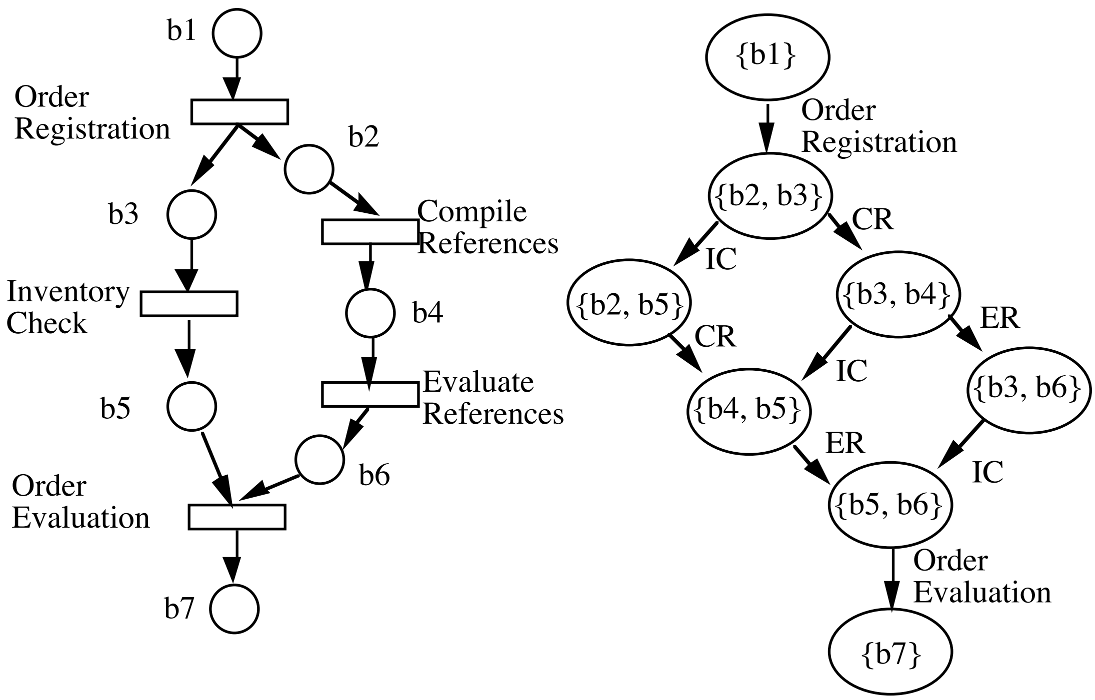

Figure 6. The initial part of the order procedure definition: both the EN System (left) and the ET System (right) are shown.

The main idea of the MWMS is to allow simple workflow definition (without affecting in the meanwhile the overall behaviours). We want the workflow its users are designing to be as simple as possible; in fact the final objective is to create a workflow model allowing its users to design their workflows even if they have little or no experience with computer science, programming, formal languages, etc. A step forward in this direction is that a user of the MWMS can design the standard steps of the workflow, i.e. the target embedded in it, disregarding any exception and/or breakdown which may occur during its execution. In fact the MWMS generates - and controls - for any definition of a network of activities a set of general exceptional paths (namely jumps, for more details see (Agostini et al., 1994c)). In general the jumps allow us to roll back, to jump forward, to execute unforeseen activities etc.

In order to discipline exceptional paths it is useful to introduce different levels of responsibilities in changes; in the MWMS an exceptional path either can be activated directly by the actor involved in the breakdown or needs preliminary negotiations with other people with respect to the difference from the standard path. We classify jumps basically depending on how much the exceptional path moves away from the standard behaviour and depending on how many people are involved in the change (for the formal definition once again see (Agostini et al., 1994c)). Therefore a simple change needs low responsibility while a more complex one needs an agreement with a higher role. We recognize four main roles related to different types of responsibility on procedures: the actors of the activities of the procedure, who are responsible for the execution of the single activities they are in charge of; the initiator, who is responsible for the correct execution of the whole procedure, i.e. she is in charge of the performance aspects of the procedure; the owner, who is the person responsible for the design aspect of the procedure; and finally the supervisor, who is responsible for figuring out the right parameters to define the procedure, i.e. she defines the right strategy in order to meet the client's requests. Some practical examples of how these roles are related to changes in the standard path are:

- to come back to the previous activity (since some information is missing); it involves just the two actors of the two activities, so the negotiation is carried out only between them;
- a backward jump of more than one step (where the actors of the intermediate activities are involved); it needs the agreement of the initiator in order to handle performance issues;
- a forward jump speeding up the execution; it involves in the negotiation both the owner and the initiator, in fact some activities will not be executed and so the owner responsible for the design must be consulted.

Summarizing the breakdowns are handled by coupling communication and negotiation phases to action phases as it occurs in real work settings, on the one hand exploiting the features of the coupling of the MWMS to the MCH, on the other underlining the skills and the responsibilities on the fulfilment of the client.

Another important issue in order to make the WMS more flexible is - besides changes affecting only a particular instance of a procedure - to allow changes in the procedure definition involving all the ongoing instances. Thanks to a simpler workflow model we can support global changes more easily; in fact it is easier to check correctness with respect to the given specification as the definition does not embed any unnecessary path. The formal definition of the global changes are currently under study; in particular MWMS will support global changes without the restart of the system, both preserving correctness with respect to the given specification and taking all instances in a consistent state. Ellis, Keddara and Rozenberg (1995) have proposed an elegant solution to this problem. With our simple workflow model, their solution has a wider application domain.

We want to underline that, thanks to the interaction of the Conversation Handler and the Workflow Management System, besides handling exceptional situations more easily, it is possible to support also those work processes needing an evolutionary definition. In fact, the MWMS allows us to generate step by step the flow of activities; i.e. when an actor finishes her activity she passes the folder to the next actor defining - through an activity template - what to do next. This implements the idea of an electronic circulation folder in a way similar to the one described in (Karbe et al., 1990). Finally let us remark that the definition of the workflow can be at an abstract level; i.e. the procedure can be started without a complete definition of the single activities refining them when more information is available. This issue has also been addressed in other systems such as Regatta (Swenson, 1993), which influenced our work.

The Architecture of the Milano System

Motivations for a New Architecture

A variety of technical reasons can be envisaged when looking at failures in using CSCW applications in real working settings. Mainly we agree with the reasons that Borenstein identifies in (Borenstein, 1992) and that are summarized in the following.

First of all, potential participants in a cooperative setting use a wide range of different environments not known in advance since the group dynamism supported by telematic networks makes the group boundaries never fixed in advance. This kind of dynamism intrinsically involves the problem of making heterogeneous environments working together, at the level of both communication standards and data structures. Moreover users can be discouraged in daily using systems not well integrated with their already existing environments, as communication, collaboration and coordination functions are always related to each other; and a system not taking that into account is not realistic or has a very partial usage. E-mail is often recognized as the most successful groupware application. People use it to communicate about ongoing cooperations; and add-on features integrated with e-mail are recognized as enriching existing ways to work.

Finally we can make some reflections about the role of end-users in installing and setting up their own working environment. More and more users boast so great a technical knowledge of computers that they prefer systems they can install and configure without relaying on remote installations that can be difficult to obtain.

Let's now come back to electronic mail. At a functional level, it has been a success, because it matches the way people work. It is asynchronous, easy to learn, and ready to be electronically stored. At a technical level it has been facing the problem of heterogeneous environments from the very beginning, as well as the problem of standards. RFC822 and MIME let millions of users exchange messages (even multimedial ones) on the Internet, by way of the SMTP (Simple Mail Transfer Protocol) that allows the messages to be exchanged by the heterogeneous hosts of the net.

Having this kind of features, e-mail is a natural candidate to be the middleware for groupware, i.e. the enabling technology for it, offering the standards that can be used as a starting point for addressing the problem of heterogeneous environments. Moreover if messages are used to carry interactive code, we have an easier integration of applications within the e-mail environment, facing in this way the second problem of integration with applications already in use to cooperate.

The Enabled Mail Model

The idea of using e-mail as enabling technology for cooperative applications is not new and is referred to under the name of 'enabled mail'; by enabled mail we mean those technologies significantly increasing the power and utility of electronic mail systems. In particular within the enabled mail family the term computational mail is used when specifically the enabling of e-mail comes from the embedding of programs within electronic mail messages (Borenstein, 1992). As we will see later, the work in this field has been carried on for many years. But only in the last period it has dealt with the standards, portability and security aspects that are crucial in this kind of environment, as we will see later when describing the environment for enabled mail application Safe-Tcl (Borenstein, 1994).

In figure 7 the model for message handling that was developed in 1979 by IFIP (International Federation for Information Processing) is depicted. This simple model can be used to characterize the enabled mail mechanisms.

First of all this model identifies three protocols involved in the messaging process:

- the protocol between the agents used locally by the users for reading the email; in the Internet community they are RFC 822 and MIME;
- the protocol between two different MTAs (Message Transfer Protocol), that is necessary to deliver the messages along the host path to the destination site; in the Internet community it is SMTP (Simple Mail transfer protocol);
- the protocol between an MTA and the user agent, used locally by the last MTA and the user agent; they are chosen locally and the most common are SMTP and POP3.

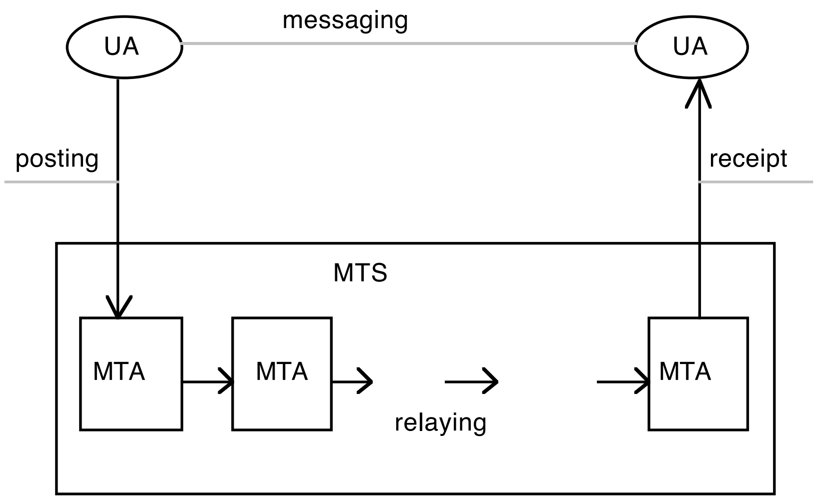

Figure 7. The IFIP model of messaging.

If we look at the life of the message three moments connected to the delivery of it can be identified. Specifically they are: delivery time (just before the message crosses the delivery slot to the user local mail environment), receipt time (after crossing the delivery slot), and activation time (when the recipient reads the message). Having such a classification, we have no interaction with the user in the first two defined phases, while it is required in the third. So examples of programs evaluated during the first two phases are respectively the sending of an acknowledgement of receipt (delivery time) and the execution of rules for moving the incoming messages in folders (receipt time). The last case comprises all the cases when an action of the user is required, for example for filling in some information to be returned to the sender.

Using such a classification we can see that existing commercial applications are always enabled with programs of the kind of receipt time. An example of an enabled mail system is Beyond Mail. It is based on the system Information Lens (Malone et al., 1986) and basically adds a rule engine to the mail agent, in such a way that users can define rules to be applied to messages and to documents routed in the system. The popularity of the rules engines depends on the fact that they can be applied in a safe way, as they just use the information that can be obtained from the semistructured fields of the messages, preserving the safety of the execution because the executed code is defined by the local user. Instead at activation time we have the interpreter using both the information contained in the message fields and the program provided by the sender. This possibility increasingly augments the potential of the enabled mail, since, for example, documents can carry with them parts of the coming execution environment, thus allowing interaction also with users not provided with the cooperative application used by the sender. Obviously the execution at activation phase reveals the security aspect and the need to have an environment clearly taking it into account.

Let's now see the Safe-Tcl environment (Borenstein, 1994) - dealing with the standards, portability and security aspects - and which features it offers to develop enabled mail applications.

The Used Environment: Safe-Tcl

The Safe-Tcl interpreter is based on an existing language called Tcl (Ousterhout, 1993) and on its graphical extension Tk. Tcl is a general purpose language, whose interpreter has been designed in order to make it easy to embed the language interpreter in applications. Starting from this Safe-Tcl wants to address the goal to be a powerful and extendible language to build applications based on the enabled mail architecture, while addressing the portability and security issues. A subset of the original Tcl primitives has been enriched adding commands to handle MIME messages; and the Tk primitives to design portable interfaces are inherited. Moreover the security issue has been carefully considered designing a particular mechanism of safe interpretation.

With respect to Tcl, Safe-Tcl is an enriched subset, in order to address both, power and security issues. Since Tcl is a general purpose language, a set of primitives has been designed to handle email oriented computations, while on the other hand a subset of the original functionality has been removed/replaced.

Of special interest is the Tcl feature to be interpreted rather than compiled, because a mechanism of extendibility has been designed thanks to it. The extendibility feature enhances Safe-Tcl with respect to a previous system like ATOMICMAIL (Borenstein, 1992), as primitives and mechanisms for the distributed maintenance and evolution of the basic set of safe primitives are provided. Basically the system can be aware of sites on the net where to look for extension scripts not found locally, and the interpreting mechanism allows us to use them in 'real time' avoiding the need to recompile the entire system.

Language Openness and Portability

Enabled mail is defined in such a way, as it uses the email mechanisms to carry not only data but also computational parts; i.e. the mailing system is enabled to execute programs inserted in the messages. This leads immediately to the problem of a standard for specifying the computational part, since in this case gateways can be provided most likely to work properly only on the data part. Safe-Tcl is a product of the Internet community, and it reflects that in being a language that exploits MIME on the front of the messaging format standard and Tcl/Tk on the front of a scripting language embeddable in applications. MIME is a real standard since the Internet community has adopted it for the messaging of multimedia mail, while Tcl/Tk is going to become a standard also outside the community. Safe-Tcl poses itself as the best candidate to become the standard in the field of enabled mail applications. It should be noted that the increase in the use of Tcl as embedded command language will increase and facilitate the way applications can communicate with each other.

Moreover it presents two other interesting features related to the interface handling and to the modality of message execution.

In regard to the interface currently the use of the X11 environment is possible, and if not, a generic interface style can be activated to provide a basic level of interaction with applications.[30]

Finally the applications need not be compiled and can also be loaded dynamically during execution; this obviously leads to greater flexibility of the environment.

So, if we compare the above system features with a broad definition of open systems: a system is open if it interoperates fully and smoothly with other systems, and if all of its communication paths are structured in such a way as to permit or encourage such interoperation, Safe-Tcl is an open system on almost all aspects.

Language Security

One of the key issues in developing an engine for enabled mail is security. In Safe-Tcl this aspect has been addressed by 'cleaning' the Tcl language (and the Tk extension) of all those commands that could be dangerous if not used appropriately. This involves commands that can activate execution of other programs (like exec or

30 Actually it is planned to port the system also on PC and Mac platforms.

source) or commands that access user's data (like gets and puts). This is the basic level of use of the language, but it is reasonable that an application extends and needs more commands than the ones contained in the basic set. For this purpose Safe-Tcl allows the use of three mechanisms:

- when installing an application implemented in Safe-Tcl, the set of new commands it provides can be declared 'harmless', basically notifying that their activation is safe for the execution environment;
- a list of FTP sites can be declared as trusted, allowing in this way to download at execution time the required code;
- programs can be authenticated/encrypted using PGP (Zimmermann, 1995), in such a way that the activation of a received program can be based on recognition of a trusted signature.

These trusted levels of execution are possible as Safe-Tcl uses a twin interpreter mechanism (see figure 8), in such a way that one of them can only execute the restricted set of commands, and for those which are allowed by one of the above mechanisms, it can request the extended interpreter for execution.

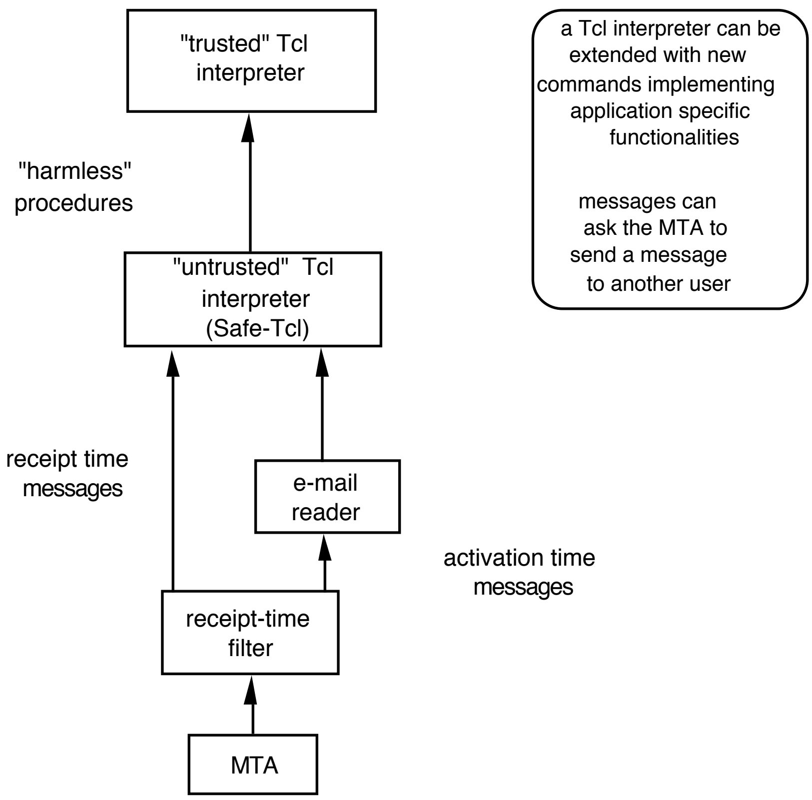

Figure 8. The twin interpreter mechanism.

The Enabled Mail Model in the Milano System

The Milano system is an *open* environment mainly devoted to being used on a WAN. The openness of applications is an aspect widely addressed for a long time. In the last years the impressive expansion of the Internet has put it even more in evidence. This is especially true considering CSCW systems, such as the Milano system, in which the user groups have not defined boundaries, are located remotely on the nets and use different platforms and systems. On the basis of these considerations, the Milano architecture has been designed in such a way as to realize the greatest degree of openness; i.e. to obtain a high level of interoperability among heterogeneous user environments (cf.. figure 9).

First of all the Milano system uses electronic mail as enabling technology; i.e. messages are used to let the users' applications interact. This can be considered the first step towards the openness since electronic mail is used by millions of people all over the world and actually is probably the most successful computer tool for working cooperatively.

Moreover, the Milano system - exploiting the potential of enabled mail offered by Safe-Tcl (Borenstein, 1994) - obtains the possibility to interact also with users not provided with the Milano application. This issue is of great importance in applications like Milano, where we cannot suppose that all actors involved in a work process have the application (e.g. someone is involved just rarely in the work process and we want to recall here that work processes cross organization boundaries).

Facing this aspect, we make the first distinction between users adhering to the enabled model, i.e. having the Safe-Tcl interpreter, and users just having e-mail .

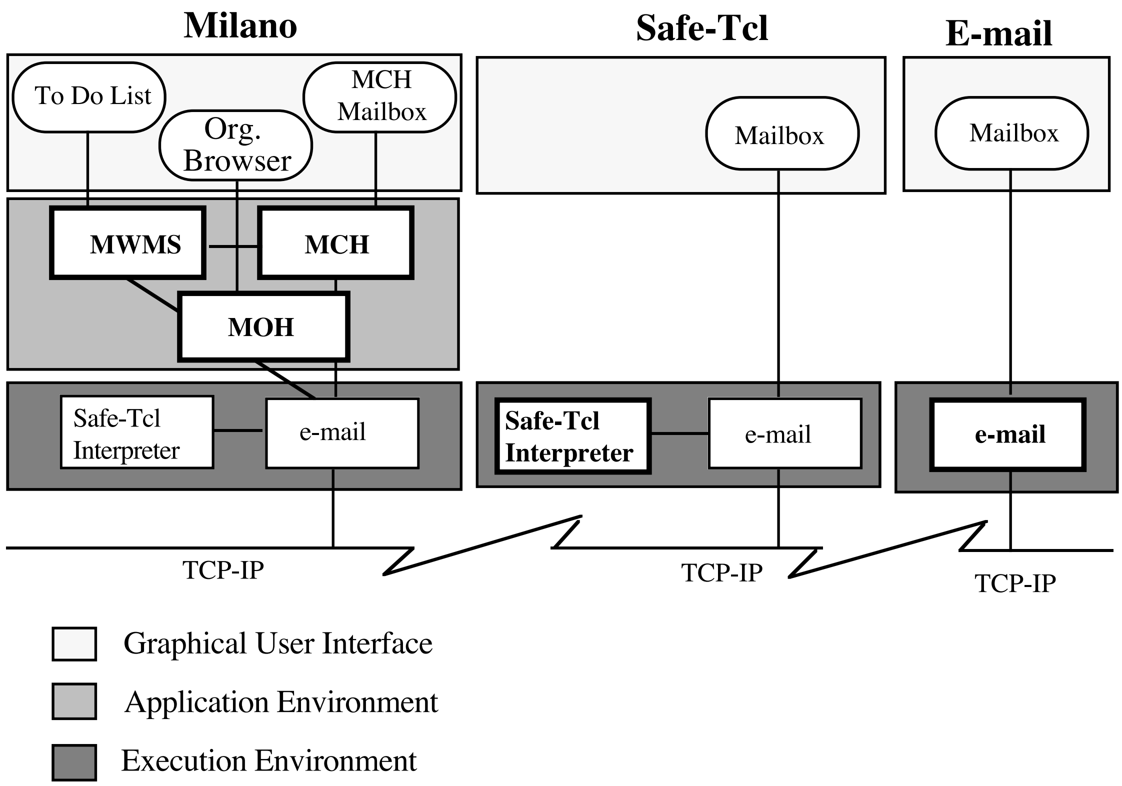

Figure 9. Heterogeneous User Environments.

A user adhering to the enabled model can still use a meaningful subset of functionalities, since Milano's messages embed data and source code that the interpreter executes on destination sites for interacting with the Milano users.

Some of the functionalities provided to this kind of users are:

- notification of activities to be performed (generated from either the MWMS or the MCH);
- visualization of the context of a procedure;
- complete handling of the flow of a procedure (calculation of the next step, routing of the involved data, etc.);
- the possibility to obtain both, missing parts of a conversation and, more generally, the state of running work processes.

We want to remark that, besides providing a lot of useful functionalities to users adhering to the enabled model, it is not important if users do not have the application. In fact no further work is added to users of Milano even if their partners (for example actors involved in procedures or participants in work processes) do not own the Milano application.

Just to add a brief example regarding the Milano Workflow Management System - similar considerations can be provided for the other components - when a new instance of a procedure is started (actually the initiator must own the application) a message is generated containing both the information regarding the procedure (for example the partial order of the activities) and the program code to be executed in the destination site (in this case the site of the performer of the first activity). Every time an activity is finished new information is automatically added to the original message before it gets sent to the performer of the next activity; instead the code part of the message remains unchanged. In summary the code part of the message knows (just from the first step) how to enrich the message with the produced data, how to calculate the next step of the procedure, how to send it, etc.

As for people just having e-mail, any program embedded in the messages cannot be executed so they will just read them; for this reason messages always contain a 'textual' part giving all relevant information, such as the description of the activity to be executed, the information to be produced and so on. In case of users having a MIME compliant reader the 'textual' part of messages is actually hypermedial. Moreover, in order to handle the flow of information, messages also embed forms for filling in some information to be returned to the person responsible.

Addressing the problem of distribution and easy installation of the application (since users can be discouraged by the difficulties in getting the system), Milano provides a mechanism for facing it. Also in this case we must distinguish between users adhering to the enabled mail model and those who do not. In the first case each message contains the code to insert the FTP site where the Milano libraries are available into the list of trusted sites (a necessary step for the reasons explained before, section 4.3.2). After this modification, the message is able to download and install (obviously without compilation) the Milano system transparently to the user. In the second case the user can still read the text part of the message containing all information regarding how to download (manually in this case) both the Safe-Tcl interpreter and the Milano application. Once again if a MIME compliant reader is available in the user's workstation the downloading of the application proves easier, since the links to the FTP site can be handled by a World Wide Web viewer.

Summarizing we can distinguish three levels of users:

- a user having the Milano application can fully use its functionalities;
- a user adhering to the enabled model can both use a meaningful subset of functionalities and easily install the application;
- a user just using electronic mail can participate reading Milano messages (in hypermedial format if a MIME compliant reader is available) and have some indications about how to retrieve both the interpreter and the application modules.

Directions of Future Research

As explained above, the Milano system is currently under development: the basic functionalities of both the Conversation Handler and of the Workflow Management System modules have been already implemented, so that its use in an open environment and its most relevant integration mechanisms have been tested, while the Organizational Handbook is still in a more primitive phase.

While the implementation is progressing, many theoretical and technical problems of the system design emerge with greater evidence, opening new research activities. Let us recall the most important of them:

(1) As we have stated above, the formal model underlying the Workflow Management System is based on a simple class of Elementary Net Systems; its 'simplicity' is relevant for both the ease of use and the small algorithmic complexity. The design of all the algorithms for the exploitation of the services delivered by the Workflow Management System to its users is an interesting application of results in theoretical computer science.

(2) We plan to use the spatial model proposed by Tom Rodden (1994) to define the proximity functions on the objects populating the Milano system. The accomplishment of this plan requires to define the criteria determining the degree of proximity between two objects, so that the presentation of the organizational context of a work process is usable and that it is feasible to switch from one work process (with its organizational context) to another one.

(3) The presentation of the user interface we are creating together with the industrial designers of Domus Academy is an elegant piece of 'Italian style' design embodying our understanding of work practices. Its development is a difficult engineering task, since the mobility of the interface provoked by the automatic agents of the Milano system can have undesired effects on the users, despite the elegance of the presentation design.

(4) We have decided to take the case study reported in (Bowers et al., 1995) as a first test bed of the Milano system: therefore we need to develop a system on top of it offering to the administrative and shopfloor personnel of the 'Establishment Printers' the same services with respect to the work accounting procedure, analyzing to which extent our solution is capable to avoid the pitfalls observed by Bowers and colleagues. We do not want to develop a real application, but only a prototype allowing us to assess the Milano system, before the completion of its development.

Other research problems will probably arise in the next phases of the development of the Milano system. To the extent that the solutions we give to them do not affect the integrity of its conception the Milano system will be a truly innovative proposal in the CSCW field.

Acknowledgements

The authors wish to thank all members of the COMIC project, in particular Wolfgang Prinz and Anja Syri from GMD. We also express our thanks to Alessandro Maioli, Roberto Tisi and Paolo Bertona for their invaluable work and the entire staff of the Cooperation Technologies Laboratory.

References

Ader, Martin, G. Lu, P. Pons, J. Monguio, L. Lopez, Giorgio De Michelis, M.Antonietta Grasso, and G. Vlondakis (1994): WooRKS, an Object Oriented Workflow System for Offices. Technical Report ITHACA, Bull S.A., T.A.O. S.A., University of Milano, and Communication and Management Systems Unit.

Agostini, Alessandra, Giorgio De Michelis, M. Antonietta Grasso, Wolfgang Prinz, and Anja Syri (1995): Contexts, Work Processes and Workspaces. C*omputer Supported Cooperative Work (CSCW). An International Journal* , (to appear).

Agostini, Alessandra, Giorgio De Michelis, Stefano Patriarca, and Renata Tinini (1994a): A Prototype of an Integrated Coordination Support System. C*omputer Supported Cooperative Work (CSCW). An International Journal* , vol. 2, no. 4, pp. 209-238.

Agostini, Alessandra, Giorgio De Michelis, M. Antonietta Grasso, and Stefano Patriarca (1994b): Reengineering a Business Process with an Innovative Workflow Management System: a Case Study. In *Collaborative Computing*, vol. 1, no. 3, pp. 163-190.

Agostini, Alessandra, Giorgio De Michelis, and Katia Petruni (1994c): Keeping Workflow Models as Simple as Possible. In *Proceedings of the Workshop on Computer-Supported Cooperative Work, Petri-Nets and related formalisms* within the 15th International Conference on Application and Theory of Petri Nets (ATPN '94), Zaragoza, Spain, June 21st, pp. 11-23. (Available on request from Dipartimento di Scienze dell'Informazione, Università di Milano, email: gdemich@hermes.dsi.unimi.it)

Borenstein, Nathaniel S. (1992): Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work. *CSCW'92. Proceedings of the Conference on Computer-Supported Cooperative Work,* Toronto, Canada, October 31-November 4, 1992. New York: ACM/SIGCHI & SIGOIS, pp. 67-74.

Borenstein, Nathaniel S. (1994): Email with a Mind of Its Own: The Safe-Tcl Language for Enabled Mail. In Proceedings of the IFIP WG 6.5 Conference on Upper Layer Protocols, Applications, and Architectures, Barcelona, Spain, June, 1994.

Bowers, John and John Churcher (1988): Local and Global Structuring of Computers Mediate Communication: Developing Linguistic Perspective On CSCW in COSMOS. *CSCW'88. Proceedings of Conference on Computer Supported Cooperative Work,* Portland, OR, September 26-28, 1988. New York: ACM Press, pp. 125-139.

Curtis, Bill, Marc I. Kellner and Jim Over (1992): Process Modelling. *Communications of the ACM,* vol. 35, no. 9, pp. 75-90.

De Michelis, Giorgio and M. Antonietta Grasso (1994): Situating conversations within the language/action perspective: the Milan Conversation Model. In R. Furuta and C. Neuwirth (eds.): *CSCW '94. Proceedings of the Conference on Computer Supported Cooperative Work,* Chapel Hill, North Carolina, October 22-26, 1994. New York: ACM Press, pp. 89-100.

De Michelis, Giorgio, M.Antonietta Grasso, and Alessandro Maioli (1995): The Milano Conversation Handler. Department of Computer Science, University of Milano.

De Michelis, Giorgio (1995): Work Processes, Organizational Structures and Cooperation Supports: Managing Complexity. In Proceedings of the 5th IFAC Symposium on Automated Systems Based on Human Skill - Joint Design of Technology and Organisation, Berlin, Germany, September 26-28, 1995.

Ehrenfeucht, A., and G. Rozenberg (1990): *Partial (Set) 2-Structures - Part 2: State Spaces of Concurrent Systems*, Acta Informatica vol. 26, pp. 343-368.

Ellis, Clarence A., R. Gibbons, and P. Morris (1979): Office Streamlining. In Naffah, N. (ed.): *Integrated Office Systems-Burotics*. Amsterdam: North-Holland Publishing Company, pp. 111-125.

Ellis, Clarence A. (1982): Office Talk-D: An Experimental Office Information System. In *Proceeding of the 1st Conference on Office Information Systems*. New York: ACM, pp. 131-140.

Ellis, Clarence A., Karim Keddara, and G. Rozenberg (1995): Dynamic Change within Workflow Systems. *COOCS'95. Proceedings of the Conference on Organizational Computing Systems*. New York: ACM, (to appear).

Jarke, Matthias, Carlos Maltzahn, and Thomas Rose (1992): Sharing Processes: Team Support in Design Repositories. *International Journal of Intelligent and Cooperative Information System*, vol. 1, no. 1.

Karbe, B., N. Ramsperger, and P. Weiss (1990): Support of Cooperative Work by Electronic Circulation Folders. *ACM Special Interest Group on Office Information System*, vol. 11, no. 2-3, pp. 109-117.

Kreifelts, T., E. Hinrichs, K.H. Klein, P. Seuffert, and G. Woetzel (1991): Experiences with the DOMINO Office Procedure System. In *Proceedings of the 2nd European Conference on CSCW*. Kluwer, Dordrecht, The Netherlands, pp.117-130.

Malone, Thomas W., Kenneth R. Grant, and F.A. Turbak (1986): The Information Lens: An Intelligent System for Information Sharing in Organizations. *CHI'86. Proceedings of the Conference on Human Factors in Computing Systems,* Boston, Massachusseth, April, 1986. New York: Association for Computing Machinery.

Malone, Thomas, Kevin Crowston, Jintae Lee and Brian Pentland (1993): Tools for inventing organizations: Toward a handbook of organizational processes. In *Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises.*, Morgantown, WV, April 20-22, 1993.

Medina-Mora, Raul, Terry Winograd, Rodrigo Flores, and Fernando Flores (1992): The Action Workflow Approach to Workflow Management Technology. In J. Turner and R. Kraut (eds.): *CSCW '92. Proceedings of Conference on Computer Supported Cooperative Work*, Toronto, Canada, October 31 to November 4, 1992. New York: ACM Press, pp. 281-297.

Nielsen, M., G. Rozenberg, and P.S. Thiagarajan (1992): *Elementary Transition Systems,* TCS, vol. 96, no. 1.

Ousterhout, John K. (1993): *TCL and the TK Toolkit*. Reading, MA: Addison Wesley, 1993.

Prinz, Wolfgang (1993): TOSCA: Providing organizational information to CSCW applications. In G. De Michelis, C. Simone and K. Schmidt (eds.): *ECSCW'93. Proceedings of the Third European Conference on Computer Supported Cooperative Work*, Milano, Italy, September 13-17, 1993. Dordrecht: Kluwer Academic, pp. 139-154.

Prinz, Wolfgang (1995): Requirements for an Organizational Information System. COMIC Document GMD-1-10.

Robinson, Mike (1993): Design for Unanticipated Use... In G. De Michelis, C. Simone and K. Schmidt (eds.): *ECSCW'93. Proceedings of the Third European Conference on Computer Supported Cooperative Work*, Milano, Italy, September 13-17, 1993. Dordrecht: Kluwer Academic, pp. 187-202.

Rodden, Tom (1994): Objects in Space, the Spatial Model and Shared Graphs, in COMIC Deliverable D4.2

Rozenberg, G., and P.S. Thiagarajan (1986): Petri Nets: basic notions, structure, behaviour. In J.W. de Bakker, W.P. de Roever and G. Rozenberg (eds.): *Current Trends in Concurrency*, LNCS 224, pp. 585-668.

Sarin, S.K., K.R. Abbott, and D.R. McCarthy (1991): A Process Model and System for Supporting Collaborative Work. In *COOCS'91. Proceedings of the Conference on Organizational Computing Systems*. New York: ACM, pp. 213-224.

Schmidt, Kjeld (1994): The Organization of Cooperative Work - Beyond the ´Leviathan´ Conception of Organization. In R. Furuta and C. Neuwirth (eds.): *CSCW '94. Proceedings of the Conference on Computer Supported Cooperative Work,* Chapel Hill, North Carolina, October 22-26, 1994. New York: ACM Press, pp. 101-112.

Suchman, Lucy A. (1987): *Plans and Situated Actions*. The problem of human-machine communication. Cambridge: Cambridge University Press.

Suchman, Lucy A., and Randall H. Trigg (1991): Understanding Practice: Video as a Medium for Reflection and Design. In J. Greenbaum and M. King (eds.): *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum, pp. 65-89.

Susani, Marco, and Giulio Cesare Solaroli (1995): Hypercard presentation of the interface for the Milano Conversation Handler. Smart Tool Laboratory, Domus Academy.

Swenson, Keith (1993): Visual Support for Reengineering Work Processes. In S. Kaplan (ed.): *COOCS'93. Proceedings of the Conference on Organizational Computing Systems*, Milpitas, California, November 1-4. New York: ACM Press, pp. 130-141.

Swenson, Keith .D., R.J. Maxwell, T. Matsumoto, B. Saghari, and K. Irwin (1994): A Business Process Environment Supporting Collaborative Planning. In *Collaborative Computing*, vol. 1, no. 1, pp. 15-34.

Winograd, Terry and Fernando Flores (1986): *Understanding Computer and Cognition: a New Foundation for Design*. Norwood, NJ: Ablex Publishing Corporation.

Zimmermann, Phil (1995): *The Official PGP User's Guide*. Cambridge, MA: MIT Press.

Chapter 10:
Seeing Information in the Organization

Tom Rodden, Jon O Brien, John Mariani and Andy Colebourne
Lancaster University, UK

Steve Benford, Dave Snowdon
Nottingham University, UK

Introduction

A large number of systems have emerged that examine the means by which different information visualizations can be used to increase access to information. Work in this area has resulted in the development of techniques to improve the user's ability to locate and use information. Many of these techniques, for example the perspective wall (Mackinlay et al., 1991), cone-trees (Robertson et al., 1991) and bead (Chalmers and Chitson, 1992) take an individualistic view of information use. They focus on improving a single user's comprehension of information and the relationship between different items of information. The implicit assumption is that users are working alone to find a piece of information and the discovery of a particular piece of information is sufficient to meet their needs. Somewhat in contrast we wish to consider the visualization of information that is explicitly intended to be shared across a number of users. In this case the information provides a focus and structure for the cooperation taking place.

This chapter briefly examines the role of shared information in cooperative work before outlining a visualization developed to support the cooperative use of information. The developed visualization seeks to provide mechanisms to allow information to be shared in a manner that encourages cooperation. To do so we focus on informing the visualization from on an understanding of the social nature of information sharing within an organization. Our starting point in informing the system is an ethnographic study of the use of documents in a technology centre.

One of the key aspects emerging from our study is the use of shared information as a means of mediating and coordinating groups of users. This coordination is reliant on the situated nature of the information within an organization and the ability to access the organizational context within which information is used. As information increasingly becomes available "on-line" there is a danger that important aspects of the context are no longer accessible and the ability to cooperatively use the information is lost. The representation of the organizational context within which information is used is central to future considerations of on-line information.

The Document Archive for a Technology Centre

The setting for our fieldwork has been a Technology Resources Centre (TRC) belonging to an international engineering and manufacturing company.[31] The TRC need to serve manufacturing parts of the company, distributed over the world, with technical advice about engineering problems, new engineering techniques, results of the tests of new products, and so on. Not surprisingly, travel is a major cost item and, as a result, the company is exploring electronic means of communication and information retrieval, both of which constitute a central aspect of the research project.

As with any organization, documents of various kinds constitute a formal exemplification of coordinating processes in work of the TRC, through memos, written instructions, plans, procedures, etc.[32] Of particular note in this case is the extremely large number of documents that are produced as part of the technical work of the Centre: technical reports, test results of equipment and materials, and so on. These are available throughout the wider company and provide a means of disseminating the technological expertise of the centre.

Recently, the TRC has introduced a pilot Document Management System (DMS) that consists of a database of OCRd electronic versions of the document archive. Distributed access to this is facilitated via an indexing system running on networked 486 PCs. This technology is part of the TRCs paper reduction drive as it migrates toward more electronic information sources. It is also intended to make access to the archive easier, more widely available within the company. In a sense, it seeks to place the knowledge of the TRC "on-line" providing immediate access for users spread across the world.

Documents within Socially Organized Work

The main purpose of the ethnographic field study was to examine the day to day working life of the engineers and managers of the TRC as a basis for the design and development of appropriate information visualization. In this chapter, we focus particularly on the various roles that documents, again of various kinds, play in the day-to-day organization of the work. In previous studies we identified on the basis

31 The names used to refer to the company and to the unit that is the focus of the research are fictitious.

32 Contracts of employment, hence the rights and the duties of particular occupational roles within the organisation are also given documentary expression. Although these are not unconnected to coordination issues, they are not our focus in this paper.

of fieldwork investigations, some general properties of documents deriving from their place within patterns of organizational work activities:

- documents as representations of organizational objects and actions
- documents as sedimentations of an organization's activities
- document access is normatively regulated
- documents have a procedural implicativeness
- the sense and import of documents relies upon local organizational knowledge
- documents as matters of inquiry and a means of access to others

What these properties emphasize is the status of documents as organizational objects which, as such, are intended to represent and display organizational activities. In other words, what documents mean, what they refer to, what they might indicate, what they constitute, what they are, has everything to do with their place within some organizational setting and its activities. What they record, namely, activities that are relevant to the organization's business. Documents represent what the activities of the organization are in a publicly available way, bearing in mind that, typically, not 'just anyone' has access to all an organization's documents. Documents are 'shared objects' and for those who know how to use them are a means of making the work activities of the organization accountable and available in various ways. This sharing is highly contingent and dependent on the context within which documents are applied.

In the work of TRC, Technical Reports are one of their main products. These tend to follow a standard format describing key technical processes, tests, equipment, specifications and so on. These documents are regarded as constitutive of what is the main product and output of TRC. They record in a publicly available manner the activities of the TRC and the experiences of previous investigations.

Of course, much of the documentation is not intended simply as a record of 'what has passed', but is often used as indications of where the work, where the organization *is* at any particular point in time and, accordingly, used as the basis for future actions and activities. Test Reports, Monthly Reports and Project Control Reports can all have a 'procedural implicativeness' for the future actions of TRC personnel. Project Proposals, for further example, set in motion various organizational processes with regard to particular forms of costing against the Centre's budget, a given Resource Area's personnel and equipment budgets amongst others, whilst the production of a Technical Report on, for example, a new testing procedure, or the calibration of a new piece of equipment , clearly implies that future actions on such tests or equipment be based on the information within these documents.

Documents as Integral to a Working Division of Labour

A point that we want to emphasise is the importance of understanding documents as integral to what we refer to as a 'working division of labour'. The division of work activities is not an idealised specification of roles and responsibilities but an ongoing mundane collection of activities that gain their sense through the 'real time, real world' actions of organizational members (Anderson et al, 1987). On this conception, the 'working division of labour' operates through an 'egological principle'; that is, from the point of view of someone immersed within a division of labour the work is seen in terms of 'horizons of relevance', as 'tasks that I do', 'getting the things I have to do, done', 'passing them on to others', 'decisions that I can take', decisions that others take', and so on. In other words the division of labour is not experienced as a totality, but as a collection of fragmented tasks which amount to the 'things I need to do now', 'things I can do later', 'things which are not my responsibility', and so forth. It is features of the 'working division of labour' that make coordination an integral aspect of any complex collaborative endeavour and an important role for documents as coordination mechanisms.

It is a commonplace to note that bureaucratic organization is organization through documents and the TRC is no different in this respect. Passing through the organization, for example, documents gather additions that make plain who has handled the document and in a number of cases what action has been taken as a result (approval for status as a Technical Document for example); in such cases the document can be seen as a 'stratified trace' of the activities of the organization and can be interrogated to this effect by those with practical knowledge of the organization itself. Thus the individual member of staff can use the document as means of representing the completion of their task and implying that it is now someone else's task to take further appropriate action. Such a system generates an order suitable for coordinating tasks across a complex division of labour, by making socially available the allocation of those tasks, and furthermore by enabling the individual member of staff to know what it is they have to do, and what can be left for other elements of the organization 'to take care of'.

Of course, what sustains this pattern is the reciprocal relationship between the various documents and the work activities which they represent. While producing documents, reading them, annotating them, signing them off, etc., is part of the work itself, what is also important is how they represent the work and, through this, are usable as a means of finding out 'where things stand', 'what is being done', 'what is yet to do', and so on. As such they constitute a public record for the staff of TRC that is 'readable' by those with the necessary local knowledge of the place, its personnel and its activities.[33]

Consider for example the way in which the placement of documents is exploited in the TRC. This is most obvious in the case of the mail trays placed on users' desks. Mail trays are arranged with an in tray on top. Mail and documents often distributed by secretarial support staff who walk by and drop them off as they pass. Staff tend to glance at or flick through in tray when they return to desk after any period away. People can readily see when they have new documents, can see if they have a lot (are busy?) if they are the only on with a certain document. Through

33 By this we do not mean that only those within TRC can read the documents for what they can indicate. Clearly, others outside the organisation would be more than capable of reading them and drawing inferences about the nature of the work and the organisation. Our point is about their readability for those 'insiders' with the practical responsibility of *using* the documents during their work.

the placement and location of documents information about the work status of individuals is made available across the organization

The properties of location central to the use of documents exploited to allow this level of coordination are not available in conventional electronic stores. The physical movement of documents from desk to desk facilitates the sequencing of tasks by locating the relevant documents ‘as someones current work’.

Knowing what the Documents Represent

The placement of documents within a setting is highly contingent on the context of use and requires considerable “local organizational knowledge” to interpret. In drawing attention to ‘local knowledge’, we are emphasising, again, the role socially organised character of documents; that is, their relationship to the day-to-day organization of activities. It involves, for example, knowing what the state of a document means for the practicalities of ’getting a project back on track’, ‘who needs to be cajoled into finding more time’, ‘knowing that X regularly overstates the time spent on a project’, and so on. For example, while it is the case that projects are planned beforehand and project review meetings are held regularly, Project Control Reports completed and filed, the day-to-day monitoring of projects also relies upon a great degree of ‘local knowledge’ as a resource that, in its turn, is built up and maintained by constant informal and ‘unplanned’ interaction between the staff of the Centre along side the use of documents.

’Informal’, ‘off-line’, ‘unofficial’ interaction underscores the relationships between staff working at TRC. They also extend to the external sites from which much of the project work originates. Such informal exchanges can be characterised in many ways as a crucial element of the expertise offered to the rest of the organization by the Technology Centre, since they typically contain the practical knowledge that often ensures successful technical work, through the use of local knowledge and tacit skills.

This use of the local knowledge surrounding the context of documents and the way in which they are used in reflected in the following extract from the fieldwork. Here two members of the TRC consider a report they are both interested in.

> "Had any more thoughts about financing the X [project referred to in proposal] ?"
>
> "Yeah - I'll have to talk to you about that ... A's passed it on to me now"
>
> "Yeah I saw earlier, I spoke to him: and checked out what he thought about using the lasers ... it's now a matter of human resources"
>
> "Yeah I'll clear this, then I'll get right back to you on this."

What is worth emphasising here is the relationship between the document as a resource and the people and activities surrounding it. While the document is the focus and reason for the surrounding interaction it itself is not the point of access. Rather, the members of the TRC are using the document as a means of coordinating their interaction with others. Knowing who has seen the document, who is currently accessing the document and finding those responsible for the production of the document is central to the support of the work.

The Space the Work Takes Place in

As an open plan office the TRC draws upon the properties of space to support and afford spontaneous informal interaction. The affordances of this space and the location of users are heavily drawn upon to support the informal interaction at the core of the work of the TRC. Users undertake their work within the office in relation to other and to the documents they use. Managing the activities of the work is as much about managing this space as the abstract activities they represent.

The location of documents and users are significant in the cooperative organization of work in that it makes activities publicly available. It allows users to infer the ordering of particular activities and who is currently dealing with information. It also promotes the possibility of interaction and cooperation. Consider the following extract from the study of the TRC.

> “R has a query re coding and accounting time-tables for some of the costing for his project - he walks up to I's desk, asks him, he's not sure - points him to notice pinned above AR's desk. (she's away) which lists the appropriate codes and time-table.
>
> R notes these down on Post-it and returns to his bay - makes a note in his diary - checks against the dates on the notice with regard to his project.
>
> Not sure about something, returns to notice, jots down another accounting code.
>
> As he returns this time:
>
> V walks down the aisle - he and R 'coincide' - JV asks how he is. 'Yeah fine ... just noting these dates down'. They reach his desk: JV stops at his desk . They talk about weather (!) V glances at R's desk ... sees project proposal form in his in tray on his desk. “

Here we see that the information is located within the office and navigating this information is navigating the office. Browsing for information is not an individual endeavour where a single user moves through an information store. Rather, looking for information involves a significant amount of social navigation akin to the processes supported by media space systems such as Cruiser (Fish, 1989), Cavecat (Mantei et al., 1991) and Rave (Chalmers and Chitson, 1992; Gaver et al., 1992)

Supporting the Context of Use

Managing the link between local informal knowledge and the information within the technical reports is a key aspect of the job of members of the TRC. In handling and dealing with requests for expert advice they need to actively apply their local knowledge to support their customers. Any developed system needs to make available the contextual information surrounding documents needed to allow members of the TRC to do their job.

The context of use is drawn upon in a number of ways within the TRC. In this section we wish to consider some of the most pertinent contextual cues used within the TRC before considering how these may be migrated toward an electronic equivalent. *These cues relate strongly to additional properties of the information.* Although presented in turn each of these different cues is closely dependant on each other and often the different contextual prompts work in tandem.

The Relations between Documents and Users

A central feature of the way in which members of the TRC exploited documents and information was to make inferences between the documents and people. These inferences relied on a number of relations

- Who had written the document
- Who had previous read it
- Who had amended it
- Who was currently responsible for it

The Relations between Documents

Similar informal inferences were drawn from the relationships and similarities between documents. Members of the TRC often considered other similar documents. These were related by

- Sharing a common topic
- Sharing an author
- Having an overlapping history of use.

The Relations between Documents and Space

One of the most important characteristics of documents exploited by members of the TRC was location. The placement of documents on desks and in specialised areas such as in-trays was used to manage relationships between documents and cooperative activities. Important spaces included

- The co-location of similar document
- The location of documents in the office
- The location of documents in relation to users desks

The central nature of each of these different contextual cues is such that they need to be mirrored in a visualization for the shared electronic repository that will replace the current paper one. This is particularly true given the organizations move toward more on-line and remote arrangements. In the following sections we outline the development of a visualization that supports many of these contextual cues to promote the shared cooperative use of this information.

Viewing the Information and its Context of Cse.

In the rest of this chapter we shall consider the visualizations that have emerged as a result of the fieldwork at the TRC. The approach we have taken is to provide a set of visual cues that support the rich set of organizational inferences central to the use of shared information in the TRC. The affordances offered by these visualizations aims to support the rich set of relationships maintained by members of the TRC.

The focus for our visualization has been the development of an electronic archive of technical documents with the TRC. The paper version of this corpus of information provides a focus for a considerable amount of the work of the TRC. An electronic equivalent will be required to provide the similar support for work across the organization.

To promote the form of work supported here two distinct forms of visualization are exploited. These different visualizations are shown along side each other and are closely linked to show the relationships between the abstract and the practical. The two areas providing a focus for visualization are

- The semantic structure of the electronic records and the relationship between documents.
- The space of the TRC and the placement of documents within that space.

Each of these different features are augmented by maintaining the relationships that exist between people and documents and allowing users to interrogate and exploit these relationships

The Semantic Structure of the Information

In this section we wish to consider the way in which we can view the relationships between documents and how these relationships can be shown to users. This feature of the visualization considers the semantic content of documents and how these can be related to each other. A number of abstract visualization of collections of information have been developed that allow views of information to be determined by the semantic content of the documents.

The two developed visualizations are termed populated information terrains and allow users to inhabit the same abstract space as the information they are interested in. The first of these visualizations uses statistical techniques to cluster information while the second adopts a more direct mapping between the attributes of the information and the position of data in space.

VR-VIBE

VR-VIBE adopts a statistical approach to represent a set of documents in a space by matching the document space against a set of queries. A query is specified by defining a number of “Points of Interest” (POIs) which contain keywords pertaining to the query. A full text search then takes place on the document space and the keywords are compared to individual documents producing a relevance score for each document and point of interest. This determines how the space will be laid out; a vector is defined giving the position of each document in space in relation to the POIs.

The documents occupy three dimensions in space and it is possible to fly through the document space. Properties of the icons representing documents in the space are used to distinguish between documents and to show selection. In particular, colour is used when a document is selected. To further emphasis selection the particular item moves position slightly.

VR-VIBE allows the user to store views into the document space and to switch between different views at will. This permits comparison of different views onto the same document space and aids in the task of navigating around the documents (if you get 'lost' you can always return to your starting point).

In the document space generated by VR-VIBE (figure 1) documents are represented by rectangular block icons while the POIs are octahedral in shape. In figure 1, query with five POIs has been defined with one of the POIs placed vertically above the other four forming a pyramid. The documents placed in the space according to the scores held in the internal tables relating documents to POIs.

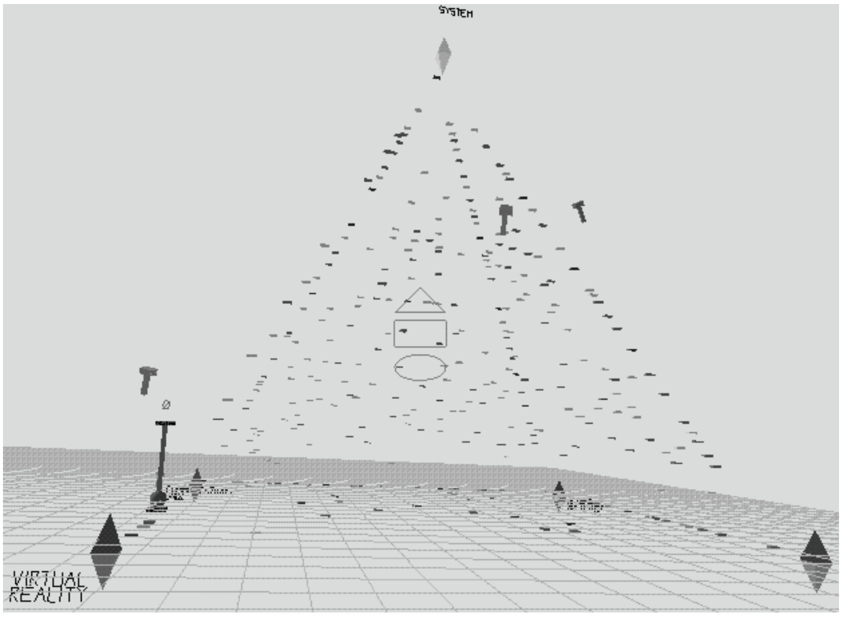

Figure 1: A query with five POIs in three dimensions

There is still a large amount of information in view and the user may want to make things clearer by highlighting the documents that are more relevant to their query. This is known as filtering. Filtering can be tailored by specifying the degree of relevance required before a document is displayed. The user tells the system to only display documents which most closely match certain search criteria.

Q- PIT

Our second approach, Q-PIT, adopts a more direct approach to the layout of information in the PIT. Q-PIT maps tuples onto graphical objects according to an extended schema notation. As an example, consider an information store which contains tuples representing exam marks within a university. The tuples include the following domains: (student, type, class) and a range of exam marks. The type identifies what degree scheme the student was registered under (single computing major, combined major, or information engineering). The class reflects what level of degree was awarded (first, second upper, second lower, third, pass, fail). For each domain in the store, we build and maintain a list of values appearing within that domain. Each domain list is then sorted into order. An attribute mapping file is created containing the following additional schema information:

```
extrinsic type x
extrinsic 300 y
extrinsic 340 z
intrinsic class shape 1st sphere 2.1 cone 2.2 cylinder 3rd hemisphere
                      pass pyramid unknown cube
```

We can see the mappings from the attributes type, course number 300 (final year project) and course number 340 onto the extrinsic (x, y, z) axes. We can associate a field and a field value with a shape; this dictates the shape of the graphical object that represents the tuple. In our simple example, we have associated the field "class" with the shape mappings, followed by a list of (field value, shape) pairs. If a field value does not appear in the mapping, the shape defaults to a cube. It is possible to have as many (field value, shape) pairs as required, with duplication of shape mappings if necessary. If the tuple has the value "2.1" for field "class", then the matching graphical object has a cone shape. Similarly, if the tuple is "3rd", the object is hemisphere shaped. A Q-PIT showing a portion of an exam result database is shown in figure 2.

A further behavioural mapping is possible with spin_speed which dictates whether the object should rotate, and if so, how quickly. In the example, we could map the value of an exam mark onto the speed. An alternative mapping for size could be to the height of the object that would reflect different students' marks in a course by the size of the object in Q-PIT.

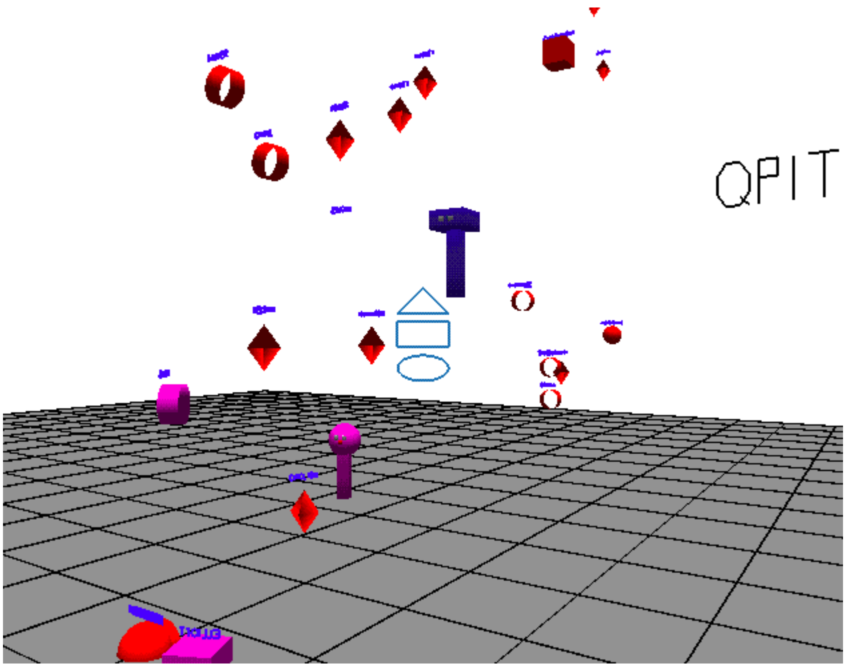

Figure 2: A Q_PIT of exam marks

Data may be manipulated in a number of ways principally in the desktop mode of use. By selecting an object the underlying record is displayed on the screen In addition, by issuing a query, all objects “hit” are automatically selected and highlighted. A user can also update the values of the underlying records by engaging in an interactive sequence of textual instructions. Once an update has been completed, not only is the underlying tuple updated, but so is the corresponding graphical object. This may entail alterations to both intrinsic and extrinsic attributes, the latter taking the form of a *smooth animation* of the object moving to the new position. We believe that the display of smoothly animated changes in position as opposed to instant repositioning to be particularly effective at providing people with peripheral awareness of changes to nearby data. Lastly, again through textual interaction, a brand new tuple can be added to the data. This is matched by the creation of a new graphical object.

Showing Documents in their Organizational Context

The visualizations of abstract structure within the document store are complemented by a more physical visualization. This physical visualization provides a document with a notion of position in space and allows users to exploit this position. The position is provided through a model of the TRC open plan office.

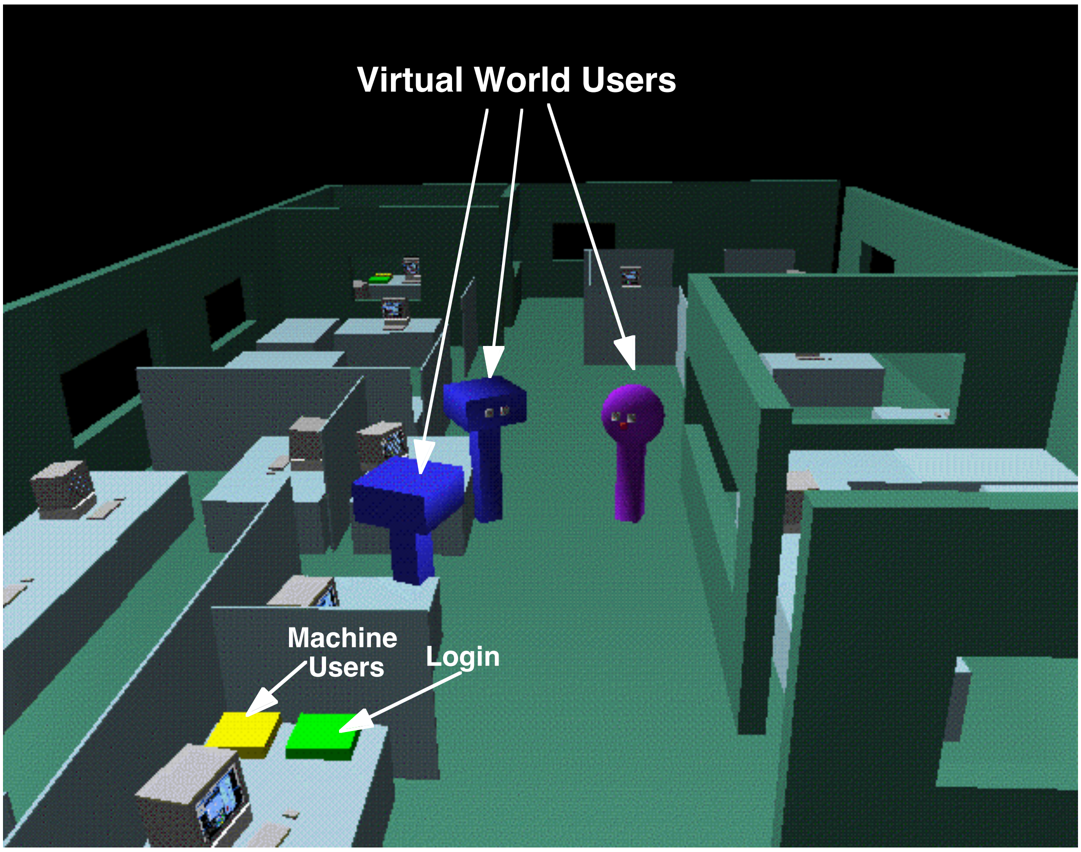

Figure 3: A meeting in the virtual world

The visualization is a facsimile of the TRC's open plan office and allows remote users to meet and share on-line technical reports. The virtual space allows users to have a "desk" where documents they are dealing with can be placed. This world model can be shared by a number of users across the network and it may be presented in either immersive or desktop mode. Figure 3 shows the office space being shared by three users. Users are embodied in the world using simple geometric shapes often with texture mapped pictures of users.

This world is also with a variety of interactive objects and applications. When these worlds are run in desktop mode we also can present the user with interfaces to applications. This allows us to provide direct access to remote users, machines and resources through the visualizations. We have also populated the world with dynamic applications that modify the state of the world. One of these focuses on making available the presence of users in the world. A simple application monitors the network to detect the activities of users on machines in the real office. We detect the presence of users on machine consoles and how long has passed since they have last been active. This information is presented to users by positioning objects representing users in front of the virtual machines (figure 4). The colour of each object is used to indicate how long has elapsed since the user was last active on the machine.

Each user object placed by the application in the virtual world is itself an interactive object. When users of the virtual world touch these objects the user's world wide web page is accessed and displayed (figure 4). This also allows direct video contact with users. The developed space allows users to coordinate their work in relation to each other and to make their presence and activities available to others.

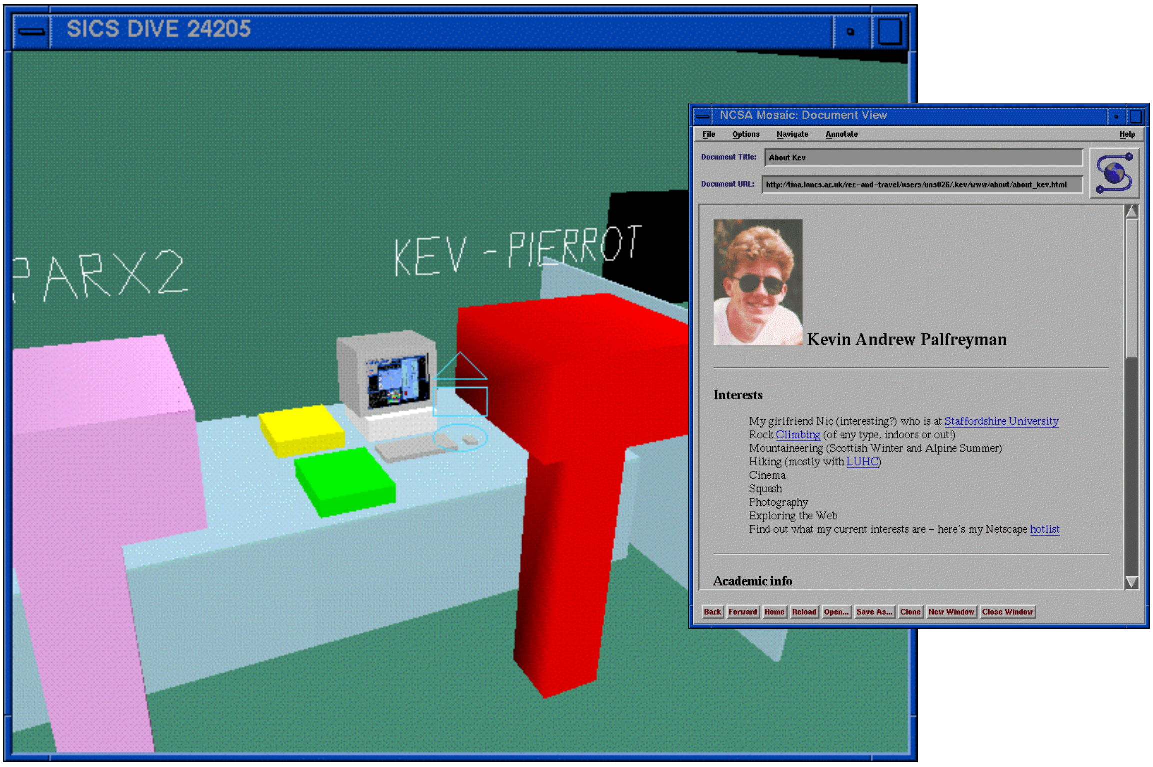

Figure 4: Accessing a users home page

Viewing the Information in Context

The different visualizations described in this chapter highlight different particular aspects of the information at the TRC. The abstract visualizations focus on the content of the information independently of its context of use. This view of information is complemented by a representation of the organizational context within which the information is used. In our case the organizational context is represented through a virtual facsimile of the open plan office at the technology research centre.

It is important to stress that these alternative visualizations co-exist and users can make use of the visualization most appropriate to the task at hand and exploit the various contextual and organizational cues offered. The availability of different information presentations and the ability of users to smoothly move between the different visualizations of the information allows users to browse and find information both in terms of its semantic content and its context of use. The representation of the physical location and the association of documents with people allow users to make inferences of the information based on the nature of the organization and the structure in place. The ability to link and query the documents in a variety of different ways allows user to make a considerable number of inferences based on both the content and organizational context of the information.

Summary and Conclusions

In this chapter we have considered a field study of a technical centre for a large manufacturing company. A central role of the centre is to act as a centre of expertise for the company. As part of this task the centre manages a corpus of technical reports that record the results of previous experiments and trials. These documents are inherently shared across the members of the TRC and much of the task at hand is in managing this sharing.

The technical reports are also drawn upon as a resource in responding to requests from across the company. They provide a point of reference and enquiry for users remote from the centre. However, the documents themselves are seldom used in isolation of the context in which they were produced. Members of the TRC often supplement the information in the technical reports with commentary and with reference to others. They often make recommendations that users contact document authors or those that have previously seen documents. Similarly, they exploit the location of documents in space to infer both the status and availability of reports.

As a result of our field study we have developed a visualization of the TRC. This visualization is distinct from previous approaches in that it aims to promote the sharing of documents across a community of users by allowing access to users and their activities through the document store. Central to this provision is a representation of the organizational context within which the information is used. The developed visualization exploits the semantic context of the documents to allow documents to be found based on their content. This semantic focus is complemented by a spatial consideration which allows their status and availability to be inferred from their position in space and their relation to users within an organization.

References

Chalmers, Matthew, and Paul Chitson: “Bead: Explorations in Information Visualization,” *Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Interface Design and Display, ACM Press, New York, 1992, pp. 330-337.

Fish, R.S.: “Cruiser: A multi-media system for social browsing,” *The ACM SIGGRAPH Video Review Supplement to Computer Graphics*, vol. 45, no. 6, 1989. - ACM Press, Baltimore, MD. Videotape.

Gaver, William, Thomas Moran, Allan MacLean, Lennart Lovstrand, Paul Dourish, Kathleen Carter, and William Buxton: “Realizing a Video Environment: EuroPARC’s RAVE System,” *Proceedings of ACM CHI’92 Conference on Human Factors in Computing Systems*, Studies of Media Supported Collaboration, 1992, pp. 27-35.

Mackinlay, Jock D., George G. Robertson, and Stuart K. Card: "The Perspective Wall: Detail and Context Smoothly Integrated," *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Information Visualization, 1991, pp. 173-179.

Mantei, M., R.M. Baecker, A.J. Sellen, W.A.S. Buxton, and T. Milligan: "Experiences in the use of a media space," *ACM SIGCHI Conference on Human Factors in Computing Systems, New Orleans*, ACM Press, 1991, pp. 203-208.

Robertson, George G., Jock D. Mackinlay, and Stuart K. Card: "Cone Trees: Animated 3D Visualizations of Hierarchical Information," *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Information Visualization, 1991, pp. 189-194.