like, it is important for smooth implementation to be able to anticipate where resistance will occur. However the main point is that end-users know their procedures better, and may help the designers to better solutions.

The initial model will consist of a full description of what actually goes on in the organisation, and what the real objectives of the management and staff are. This initial model will then stand as the basis for the design process, which will lead to the creation of a model for the proposed situation when the PSS system is introduced. This model in turn will consist of two parts: one of the environment in which the PSS system will run, and one of the PSS system itself. The latter will be in a form which permits the code for the system to be generated directly from it. Given that the system will be constantly evolving, it would be useful in small systems, and essential in large ones, always to have an up-to-date model for purposes of implementing and keeping track of changes.

Work is continuing under the IOPT program on forms for models. Originally PML was, as its name suggests, intended to be a language in which to model the processes, and its main constructs - roles, activities, entities, and interactions - reflect this. However, the requirement of executability has led to scripts written in the language being very difficult for humans to understand.

5. Conclusion

PSS attempts to co-ordinate routine co-operative work. In order to do this it embodies a description of those work processes. However, these processes are in essence normative and subject to a process of dialectical development. PSS provides a framework within which systems can be built which tightly prescribe or loosely guide as required, and which can develop and grow. But there is still much work to be done on tools and methodologies to help system designers in the creation of suitable structures.

6. References

[1] L. Bannon and K. Schmidt. CSCW: Four Characters in Search of a Context, in Proceedings of the First European Conference on Computer Supported Cooperative Work 1989

[2] T. Winograd. A Language/Action Perspective on the Design of Cooperative Work, in I.Grief(ed) Computer Supported Cooperative Work. Morgan Kaufmann 1988

[3] J. Maresh and D.Wastell. Process Modelling and CSCW: an Application of IPSE Technology to Medical Office Work, in INTERACT ‘90. IFIP 1990

[4] E. Gerson and S. Star. Analyzing Due Process in the Workplace, in ACM Transactions on Office Information Systems Vol.4, No.3, July 1986, Pages 257-270.

[5] B. Warboys . Proceedings of International Conference on Systems Development Environments, Berlin 1989.

[6]B. Shiel. Coping with Complexity, in Office: Technology and People Vol. 1. 1983

4. Designing PSS Systems

Early attempts to automate the office failed for two main reasons: firstly because the hardware was not sufficiently sophisticated, and secondly they adopted too simple a model of how offices operate. Many authors have decried "the traditional bureaucratic model" of the office (e.g. [1], [6]), which assumes that everything happens according to well-defined procedures, and neglects the role of informal networks and the inherent complexity of supposedly routine tasks. Whilst we would certainly not wish to suggest that informal networks and ad hoc procedures do not play a large part in the successful running of most organisations, and that the procedural structures may have evolved rather than been planned, those structures do exist in almost all organisations. They may be planned, and they may be restructured in a planned way. The challenge for IPSE system designers is to build systems which incorporate the formal structure, but allow and if possible facilitate the informal. What PSS provides is a platform on which to build systems. The detailed characteristics of the eventual systems built on it will depend on the system designers. The work currently going on in the IOPT project includes investigation into methodologies and tools to aid system designers.

In order to produce a PSS application that will effectively support the processes of an organisation it is necessary to have a thorough understanding of what those processes are and how they work. The method of gaining this understanding, and ensuring that it is shared by all those involved in the development, is by the construction of a model (or set of models). A PSS system could not be built from a set of functional requirements alone, nor even with just the addition of a set of user requirements. It is envisaged that a PSS system will lay the framework for all that happens in its target organisational area, and so the design of the whole socio-technical system is at stake.

The first requirement in building a PSS system is thus the construction of a sufficiently rich model of the target processes. This model aids reasoning about those processes. It helps to highlight gaps and inconsistencies in designers knowledge. It provides a common ground on which to base discussions between system designers and the various users of the system. These people will all have different perspectives on what goes on in the organisation, and different values will be given by them to certain parts of the process, according to their different tasks and objectives.

It is most important that the model should capture the views of all concerned with the processes, and reconcile any differences. The reconciliation process needs to identify why those differences occur. In many organisations there are procedures manuals, but there can be great variations in how closely they are followed. An early pilot application of PSS was built largely from the procedures manual of the target organisation, in consultation with the manager. When the application was presented to the main end users they quickly pointed out ways in which it conflicted with their actual work practices, and would be less efficient than previously.

There may be many more or less good reasons why standard procedures are not strictly adhered to within an organisation, but even if they are 'bad' reasons (from the point of view of the efficient running of the organisation), it is still important that they be understood. Failure to involve end-users in the design of any system is bad practice, in the case of CSCW systems it can be disastrous, as they are so dependent for their success on the goodwill of the end user. Even if in the end management is determined to press on staff a solution that some will not

3. Integrated Process Support Environments (IPSE)

An IPSE is an environment in which a computer system supports work by actively co-ordinating the work of a number of co-operating actors, making sure that resources and tools are available at the right place and the right time, and automating the more straightforward and laborious procedures [3]. Originally IPSEs were aimed at software engineering processes but they have much wider application. IPSE 2.5 refers to an IPSE built by a consortium , involving STL, Manchester University and others under the UK government's Alvey programme. This technology is now known as PSS (Process Support System) and is being further developed under the Introduction of Process Technology (IOPT) project, led by STL, also funded by the UK Department of Trade and Industry. PSS provides a programming language called PML in which, in principle, processes can be formally described. Currently, our convention is to describe processes in terms of entities(e.g. a document), actions (e.g. type a document), roles (e.g. secretary) and interactions, which express the connections between roles (e.g. send document to boss).

At the heart of PSS is the Process Control Engine (PCE). This is programmed in PML through which it is, so to speak, made cognisant of the processes to be supported. The PCE provides the appropriate working environment to each user of the system. It informs them of their role in the process, the state of progress on the activities in which they are involved, allows them to select an appropriate activity on which to work, providing access to the requisite objects (e.g. documents), and tools (e.g a word processor). PSS can integrate the applications that the organisation already uses and automate the passing of data between them if necessary. Once the user has finished their task, the results may be passed automatically to the due recipients.

Large computer systems tend to stand in the way of organisational change because they themselves are difficult to change, and other work processes have to accommodate them. A PSS system is potentially very invasive, in that it is envisaged that it would spread to incorporate all the procedures of an organisation, and these procedures must be able to evolve if the organisation is to remain successful. Therefore it is particularly important that it should be easy to change a PSS system.

This was always recognised by the designers of PSS, and hence facilities for change were built into its structure. The most important point is that changes can be made to the system whilst it is running. Also changes can be strictly local. This means, for instance, that a manager may make changes internal to their department without anyone outside that department being affected, or even necessarily knowing that the changes have occurred.

Although PSS as it stands contains the basic facilities necessary, there is more work to be done to facilitate the building of systems which can evolve, and this is being undertaken as part of the IOPT project. It must be easy for the users to make desired changes. They will also need to be able to keep track of how things work, and calculate the direct and indirect consequences of any changes they make. For this they will require tools, such as a graphical representation of the target processes, which can be manipulated to produce the changes. A system of permissions will also need to be built in, so that people can only make changes in areas for which they are responsible.

Some work has been carried out on the creation of macros which can be built into PSS systems to facilitate change. One of the best developed of these is the Process Model for Management Support (PMMS). This is described in Warboys [5].

to support ad hoc co-ordination by Winograd with his 'Co-ordinator' [2]. IPSE technology, on the other hand, supports co-ordination of routine tasks.

2. The concept of Process

The objective of this paper is to discuss a form of technology, the Integrated Process Support Environment (IPSE) which, as the name implies has been developed to automate organisational processes and has been advocated as a base technology for constructing CSCW systems [3]. In this section we discuss our view of the nature of "Process" as a prelude to a discussion of the use of IPSE technology for implementing CSCW systems.

The idea of Process is closely associated with the larger concept of organisations as bureaucracies. Bureaucratic organisation is characterised by the use of explicit rules and procedures to provide routine co-ordination within the organisation. The idea of Process refers to a routinised way of carrying out some desired activity. Often such processes are complex and involve the coordination of the activities of many individuals. Frequently they fail or are inexpeditiously executed, which is why computer support might be beneficial. The idea of Process is thus that of a consciously designed procedure involving people and materials designed to accomplished a well-defined objective.

An important issue is the relationship between processes and reality. Our definition of Process sees processes as normative, that is a process defines how work ought to be done. It does not, of course, determine what people actually do. At one extreme, rules may be followed slavishly, at the other, there may be complete laissez-faire. In the middle, there is a broad range which we believe to represent the typical situation, in which formal procedures provide a general normative structure within which work takes place. Workers cannot and do not normally ignore procedures, whatever their views about them. They generally work within the rules; in a real sense they make the rules work. Because procedures are prescriptive they should not be seen as purely constraining. Procedures allow people to do things; they are enabling as well as constraining. By sanctioning a particular method, they also provide some dependendable guarantee that desired ends will be achieved.

The relationship between procedure and practice is thus not a simple deterministic one: the implementation of rules requires interpretation and negotiation. Indeed working out how to apply the rules, keeping them as it were in good repair is a key part of office work. As we have said, people make the rules work. In our view, therefore the relation between rules and practice is dialectical: formal procedures shape but ultimately do not determine practice; practical experience in turn can lead to re-interpretation or reformulation of procedures as contradictions emerge. Practice is invariably a compromise between procedure and exigency. This point is absolutely crucial. Procedure manuals are an idealised normative view. No formal description, however, can ever completely cover the contingent, ever-varying character of real work. The unanticipated always occurs, requiring resourcefulness and ingenuity. There are always short cuts. The work of applying the rules, of getting the work done despite the rules, is sometimes referred to as articulation. "Every real world system requires articulation to deal with unanticipated contingencies: articulation resolves these inconsistencies by packaging a compromise that gets the job done...the results of this work appear as workarounds or kludges...."[4].

IPSE technology as a basis for implementing CSCW systems

P. R. White and D. G. Wastell

Department of Computer Science, University of Manchester, Oxford Road, Manchester, M13 9PL, U.K.

Keywords: CSCW, Process, IPSE technology, system design, office automation

Abstract

The Process Support System (PSS), which was developed under the Alvey IPSE 2.5 project, provides a framework for building cooperative work systems. This paper discusses the theoretical underpinnings of PSS, in particular the concept of Process and the nature of Processes as subject to dialectical change. PSS as it stands provides a basic platform for system building. The challenges facing PSS designers, and their requirements for tools and design methodologies, are discussed.

1. Computer-supported cooperative work

Bannon and Schmidt [1] have defined 'co-operative work' as "the general and neutral designation of multiple persons working together to produce a product or service". Others have attempted to produce a more restricted definition, or to substitute the term 'co-ordinated work' for the wide definition. In this paper we take the two terms as interchangeable. However most work is co-operative in this sense, and hence most computer systems could be said to aid or support co-operative working to some extent.

What then is the distinctive feature that unifies the field of CSCW? We would argue that it is a question of focus. CSCW studies address the question of how co-ordination or co-operation is achieved, and how computer systems may be used to improve this process. CSCW systems are attempts to provide practical solutions to these problems. In practice, such systems typically address a narrow co-ordination problem domain, such as co-authorship of documents, or support for group decision-making

Co-ordination occurs over varying timescales. We may make a division into routine and non-routine co-ordination. By the former we mean co-ordination of regular events, e.g. the handling of invoices. By the latter we mean co-ordination for an ad hoc task, say an irregular meeting. The result of co-ordination of routine activities is a procedure or process which is repeatedly followed. Sets of procedures structure the work that is carried out in all organisations. Within this structure occur all sorts of ad hoc co-ordination. Both types of co-ordination, and the relation between them, seem fruitful for study. An attempt has been made