[8] E. Wynn. Office Conversation as an Information Medium. Ph.D thesis, Univ. of Calif, Berkeley, Calif., 1979.

[9] C. A. Ellis and G. J. Nutt. Office Information Systems and Computer Science. In I. Greif (ed) Computer Supported Cooperative Work. Morgan Kaufmann 1988.

[10] T. Winograd. A Language/Action Perspective on the Design of Cooperative Work. In I. Greif (ed) Computer Supported Cooperative Work. Morgan Kaufmann 1988.

[11] S. Sluizer and P. Cashman. XCP: an Experimental Tool for Supporting Office Procedures, in Proceedings of the 1st International Conference on Office Automation. IEEE 1984

[12] H. Smith, P. Hennessy, and G. Lunt. The Activity Model Environment: an Object-Oriented Framework for Describing Organisational Communication, in Proceedings of the First European Conference on Computer Supported Cooperative Work. 1989

[13] C. Tully. Introduction to the Proceedings of the 4th International Software Process Workshop. ACM 1989

[14] IT-STARTS Developers' Guide. NCC 1988.

[15]S. Greenspan. Requirements Modeling: a Knowledge Representation Approach to Software Requirements Definition. Ph.D thesis, Univ. of Toronto. 1984

[16] S. Benford. Requirements of Activity Management, in Proceedings of the First European Conference on Computer Supported Cooperative Work. 1989

[17] P. Checkland. Systems Thinking, Systems Practice. Wiley.1981

7.2 The Way Forward at Manchester

7.2.1 The role of fieldwork

It is the intention that case studies will become the dominant activity at Manchester over the next six months. We wish to gain practical experience of capturing and modelling work processes in a variety of situations. Fieldwork will be complementary to our theoretical investigations. Theories and methods will be built on what we learn in the field. It also provides us with an opportunity to test out and refine these theories and methods.

7.2.2 The development of a method

There is a need to bring together organisation theory with systems analysis techniques and modelling methods in order to develop the analysis framework described in 6.6 above. Soft systems methods are being examined as a possible initial way of capturing and ordering information about problem situations.

The examination of the suitability different diagramming methods, such as Petri nets, will be undertaken soon. Available diagramming tools will also be reviewed.

8. 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] G. Morgan. Images of Organization. Sage. 1986

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

[4] L. Suchman. Plans and Situated Actions. Cambridge University Press. 1987

[5] C. R. Franz and D. Robey. An Investigation of User-Led System Design: Rational and Political Perspectives. Communications of the ACM Vol. 27 No 12, 1984

[6] 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.

[7] P. Checkland. Soft Systems Methodology. In J. Rosenhead (ed) Rational Analysis for a Problematic World. Wiley 1989.

6.6.2 Forms for Models

There is a problem in dealing with the potentially vast amount of information generated in modelling processes in such a way that it remains human understandable. There is a need to break down the overall representation into manageable chunks, but still maintain the links. Obviously this suggests computer-based tools for storage and manipulation of the information**.** Graphical representations are the clearest for human understanding. Ideally they should be machine-understandable as well, so that analyses can be performed, and what-if scenarios enacted.

7 Conclusions

7.1 General

Wynn recommended in 1979 that “the technological revolution already underway be accompanied by a correspondingly sophisticated understanding of the environment to be transformed”[8]. With PSS the ‘technological revolution’ continues, but we are still in need of the ‘sophisticated understanding’. If PSS is not simply going to automate the problem, or automate someone’s misunderstanding of the problem, then new methods of understanding and analysing the problem situation are necessary.

A general understanding of the nature of organisations must underpin the method. It has been argued above, following Greenspan, that we need to model the requirements situation as a whole, before attempting to model a solution. However, as indicated in 6.6.1 above we need to look at a wider range of factors than has previously been the case. In particular there is a need to include the *values* of processes and actions in terms of the goals, objectives, strategies and interests of those involved.

Ways must also be found of taking account of the importance of informal interactions, communications and networks. We are faced here by something of a paradox - if such interactions can be formally described, then they can be formalised in practice. Some will not be susceptible to detailed description, but their function and importance need to be assessed.

As regards possible contributory elements to a design method, the soft systems methods developed by Checkland [17] seems to offer the best starting point in terms of an initial approach to developing models of the problem situation. It has the necessary flexibility, and breadth of scope, and very importantly emphasises the need for the analyst to work with the users and take account of differing perspectives.

rather than something decomposable into other roles or atomic actions. It might be better to use the concept of *tasks* and *subtasks.*

Standard methods of modelling processes tend to deal with WHAT happens but to be weak in portraying WHY it happens. If the intention is to analyse and improve processes then it is necessary to understand why activities are happening, This involves looking at such things as:

- goals - long-term goals of the organisation or department
- objectives - shorter term, more closely related to actual activities
- global and local strategies, policies, procedures and standards
- how the interests of organisational parts coincide or conflict
- how conflicts are resolved

Another area that needs to be addressed is the relative importance of different processes and actions, or the same process when it is dealing with, say, different clients. Certain actions may be infrequent, but be crucial to the organisation because, for instance, a great deal of money is at stake. Other processes may be relatively unimportant in each particular instance, but constitute the 'bread and butter' of the organisation. Thus there is a need to look at:

- the bases for prioritisation of actions
- importance of actions, why and in what way they are important,
- how time constraints are handled

Also important is the staff structure, and the way it relates to the processes to be performed. Any change in processes will mean a change in people's jobs, so it is necessary to work out the acceptability of such changes by examining areas such as:

- responsibilities
- authority
- staff profiles, especially skills
- career paths

From sections 2.4 and 2.5 it is apparent that there is a need to account for:

- how articulation is accomplished
- informal networks
- informal interactions

Finally, as Benford [16] points out, there is a need to look at not just internal structure of activities, but also the relation between them and the external world.

It would be easy to get to a point of recommending the creation of such a rich model that the modelling task would never be completed in practice, let alone a PSS system built. Obviously it is important to prioritise, both generally and at the time of modelling.

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. 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.

6.5 Process modelling to produce successful PSS systems

The first requirement in building a PSS system is thus the construction of a sufficiently rich model of the target processes. This model would aid reasoning about those processes. It would help to highlight gaps and inconsistencies in designers knowledge. It would provide 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.

The initial model should 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 would lead to the creation of a model for the proposed situation when the PSS system is introduced. This model in turn would consist of two parts: one of the environment in which the PSS system will run, and one of the PSS system itself. The latter would ideally be in a form which permitted 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.

6.6 Analysis Framework

6.6.1 Scope of models

From the areas reviewed above, particularly in Section 2, it is apparent that the description of the problem situation should ideally be very wide-ranging. The core would be the formal processes, that is, routinely performed sequences of actions, as approached by PML, ICN, SCOOP, and other modelling systems. These focus on *actions*, the links between actions such as *interactions* and the *conditions* for actions, and the *entities* or *data* or *documents* involved in these actions. This information can be portrayed graphically in Petri nets, Role Activity Diagrams, or other diagramming formalisms such as that used by ICN.

One possible deficiency with all of these lies in grouping actions. Actions are not only related through sequence and dependency, they can usually be decomposed into sub-actions that make up the action. PML uses roles to do this, but at times this strains the generally understood meaning of role. *Roles* are usually seen as more general, comprising work of a certain type, or a certain range of responsibilities

6.3.5 Requirements of modelling forms

The required properties of a model obviously depend on the proposed uses of the model. Ellis and Nutt [9] suggest the following properties:

Mathematical tractability

Simplicity - so that the model is easy to understand, especially by the participants in the processes.

Extensibility- to allow theoretical analysis, simulation, and implementation

Greenspan notes that “Languages for requirements modelling must offer facilities for the representation and organization of knowledge into a coherent structure that is understandable by its designer(s), by programmers building systems based on a given requirements specification, and, ultimately, by end users of these systems”

6.4 “Requirements” analysis in the development of PSS systems

There are two main ways in which computer systems fail, internally where the computer system fails to operate in the way it was designed and intended to operate, and externally, where it fails to fit into its intended place in its environment in the way its users hoped. The latter problem is a far greater one. It is usually described as “failing to meet requirements” and the usual way of trying to avoid it consists of a more thorough gathering of requirements, extending from functional requirements to “user requirements”.

The whole traditional idea of requirements is based on a functional view of computer systems, that we want the system to do X, Y, and Z. These are our requirements of it. As systems have become more sophisticated and flexible the ‘hard’ functions that they provide have become relatively less important, as against considerations of how easy the system is to use, how it ‘fits’ into its organisational context. The ‘how’ becomes as or more important than the ‘what’. In order to maximise organisational effectiveness it is no longer sufficient to look at computer systems simply as applications to perform narrow tasks or solve particular problems, we must take a holistic view of the socio-technical system and aim for synergy.

This necessarily affects the form of the software development process. Computer systems are moving from the position of being tools used for well-defined and limited purposes to being highly-integrated parts of the overall system of the organisation, indeed to a situation where people may be viewed as “process inhabitants” in a system largely computer-driven. It is not sufficient to design such a computer system in isolation. It has to be integrated into its context from the outset.

As a general principle, the more thoroughly a system designer understands the environment in which his system is going to run, the better “fit” there is likely to be between his system and its environment. The requirement for good “fit” is far higher with a PSS system than any ordinary application or tool that has a definite job WITHIN the processes of an organisation. We envisage PSS as ubiquitous.The introduction of any computer system changes the processes of the organisation into which it is introduced. CSCW systems such as PSS constitute a particularly pervasive intervention.

- A model maps to that which it represents.

It is most important to distinguish between the model and the system it models. The nature of PSS tends to blur this distinction, especially for those involved in building systems (or “models” as they are usually, confusingly, known). The designer looks at what goes on in the organisation, then goes through a design process, then attempts to write a PML program which will run successfully. They may well think that because they started by looking at the organisation and ended up with code written in a “modelling” language, which when executed is commonly known as a “model”, that they have modelled some part of the organisation. But if the objective of the designer is to produce a system to be installed in the organisation and used to control processes, then they will want to provide benefits to that organisation, and so they will be making ‘improvements’ during the design process.

There will be a tendency to premature, creeping improvements. If the designer doesn’t fully understand why processes are as they are, or in fact WHAT they are in practice as opposed to in the procedure manuals, then “improvements” are liable to be anything but.

6.3.3 The problem situation and the solution system

Greenspan, whose Requirements Modelling Language laid the basis for PML, says “It is important to distinguish whether a language for modelling is meant for describing the problem situation (“the world”) or a solution system.”[15]How are we to regard PML in the light of this? If Greenspan is correct then it is necessary to make a choice, or a split, in the development of PML.

A wider, and possibly more important question concerns the modelling activity rather than the language in which it is carried out: it is important in the activity of modelling to distinguish whether one is modelling the problem situation or a solution system.

6.3.4 The right type of process modelling for the job

Ellis and Nutt contend that “there are compelling arguments in favor of analytic modelling”, and that “the office of the future will need to lean heavily on modelling and theoretical analysis”. By this they mean modelling the problem situation, in order to understand and analyse it. The models would not necessarily be part of the development process of a computer system, although they could provide a basis for such a development.

When it comes to designing a system to support process it is first necessary to model requirements. Greenspan [15] suggests that: “instead of describing requirements in terms of what is in the system, we are promoting the idea that the world should first be described directly, in terms that are natural for thinking about the world.” Similarly Benford [16] says “In order to design or configure new activities, information describing the behaviour of current activities must be gathered and analysed.”

procedures"[9]. Likewise SCOOP began by representing office procedures as Petri nets, and ICN was developed "to describe offices to managers and to analyse the office"[9]. All of these are attempts to find ways of modelling processes.

With PSS these two strands have come together. However the thrust of the development of PSS, as in the case of Officetalk or SCOOP, is towards producing systems which will actually support procedures in everyday use, rather than simply to model them. The former is obviously a worthwhile objective, but it is a different objective from the latter, and the development of tools (in this case PML and the PCE) to advance towards one goal does not necessarily bring the other closer.

Process modelling has also been seen as part of the systems analysis and design process. The IT-STARTS Developers Guide[14] identifies process modelling as the complement of data modelling. It states that "Process modelling is a representation of the activities, functions and processes performed by a system upon the data defined in an appropriate Data Model." It suggests the use of the following methods:

- data flow diagrams
- functional decomposition
- event modelling
- state transition diagrams
- structure charts
- entity life histories

6.3 Why Model?

6.3.1 Reasons for Modelling

There are several reasons for modelling processes:

- to understand and reason about the process
- to aid communication about the process, thus helping in the reconciliation of different views of the process
- to analyse the process, and determine ways in which the process may be improved

These are general reasons for modelling processes, and do not in themselves presuppose that any computerised system to support the processes will be built. However, another reason for process modelling is:

- to form a basis on which to specify systems to support the process.

6.3.2 The model and its referent

There are two important aspects of models to keep in mind:

- A model is a representation of something exterior to itself

It may be better for quality if information concerning the processes is spread widely rather than simply channelled up the management hierarchy. This would tend to make users more rather than less knowledgeable about what goes on, and more rather than less involved in it.

5 The need for PSS systems to grow and evolve

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.

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. 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

6 Process Modelling

6.1. Introduction

Process Modelling is increasingly being seen as an important field, so there is a need to place PSS in relation to this field. Does PSS answer a need for a method of Process Modelling, or is PSS mainly a means of providing process support, and process modelling only an activity in the process of designing a PSS system? In order to answer this question it will be necessary to clarify our terms.

6.2 What is meant by the term “Process Modelling”?

The term “Process modelling” is used very loosely. It originally referred to the modelling of the software development process, as part of the attempt to “understand and compare software processes, evaluate, and reason about them, and design and replicate (reuse) better ones.” [13] However it was soon postulated that other processes than the software one could be modelled, and “modelling and programming the software process becomes an experimental test-bed for modelling and programming human-computer activity in general.” [13]

Quite separately from this, and pre-dating it by some years, there has grown up a field of work examining and trying to formalise office procedures. “The Officetalk-Zero effort began in late 1976 as a study of languages for expressing office

4 Quality, responsibility, and autonomy.

There is a growing realisation amongst the more successful organisations of the importance of quality, and an associated commitment to total quality management. The main emphasis at present is on ensuring high quality of product by means of high quality of process. This in turn is ensured largely by the enforcement of standards and conformance to a specified process. However, it has been found in practice that where attempts have been made to lay down in procedures manuals detailed instructions for the vast array of possible situations the manuals become so large that (a) they are not followed because it is too time-consuming to look them up, and (b) they become difficult to update as procedures change. Evidently PSS can provide great benefits in the area of conformance to process.

However, this aspect of total quality management can, if due care is not taken, run counter to another aspect. This is the idea that quality ultimately depends on the commitment to quality of the people involved in the process. Tight rules and monitoring will never work as well as commitment for two reasons: (a) if staff are alienated from the work process they will find ways to disrupt it, as has been repeatedly shown in industrial processes, and (b) so much of office work is non-routine or has non-routine elements, and people's innate flexibility and creativity can handle this far more effectively than formalised rules. The more willingly people do a job the more likely it is to be good quality. Thus it is imperative in designing PSS systems that conformance to process is not forced upon the users in a way likely to alienate them, but rather appears as a guide which helps them to conform.

If staff are not alienated from their work then they will:

- want to do a good job
- want to be involved in the production of a good product
- want to be efficient

as these things all help to provide job satisfaction.

The more responsible people feel about the outcome of their work the easier it is to involve them in the pursuit of improved quality. Responsibility comes with autonomy - the scope for making choices. If PSS enforces conformance to process in a rigid way then it will detract from the sense of autonomy felt by the users, especially if they do not agree with the way the process operates. This would be caused if, for instance, they see any sub-optimality in the processes thus enforced (and any rigidly-defined procedure will be sub-optimal in a certain percentage of cases).

There is another related question concerning responsibility. As Ellis and Nutt [9] point out, in any office system "some person or process should have responsibility for each transaction that enters the system". PSS can help to clarify users' responsibilities through its role and action agendas. It can also help managers keep track of the work being done by their subordinates. However, this feature may need to be used with care in the design of PSS systems, for if staff feel that managers are always looking over their shoulders it may be counter-productive.

3.5 Information Control Nets

Information control nets (ICN) were developed by Ellis, Nutt and others at Xerox PARC with the intention that they be “used both to describe offices to managers and to analyse the office”.

An ICN is a diagrammatic method of representing office procedures as a set of activities, a set of repositories, and a set of functional mappings between these elements, backed by a formal definition. An ICN defines an office as a set of related procedures. Each procedure consists of a set of activities connected by precedence constraints. ‘Repositories’ contain needed information.

The most obvious difference to PSS is that there is no concept of role, or linking people or resources (other than data) to the procedures - it is a bare functional description.

3.6 Activity Model Environment

Smith, Hennessy, and Lunt [12] describe their Activity Model Environment as “a framework for modelling organisational processes and a prototype execution environment”. It is an object oriented tool to explore models of organisations.

It has a conceptual framework formed of eight components:

- activities - to achieve a common goal, and define the basic work processes of an organisation
- people
- roles - group of duties corresponding to job title
- workspaces - conceptual work areas containing resources for particular roles e.g in-tray
- messages - exist for lifetime of activity may be document, memo etc.
- information units(iunits)- atomic information objects, comprising a name, one or more fields, and completion rules
- rules - define and constrain behaviour of other components under particular conditions. invoked under set conditions to cause set actions
- functions -atomic operations carried out by roles and messages as part of an activity, (cannot be partially carried out)

It is essentially a data-driven system. It is interesting that the designers considered it best to “associate the responsibility for activity co-ordination with specific role instances rather than abstract entities”, so that users are ultimately responsible.

be added in a PSS version, such as providing prioritisation of the activities of an individual worker according to the state of related processes.

3.3 XCP

Another system similar to Officetalk is XCP, developed by Sluizer and Cashman at DEC. They describe XCP as “an experimental coordinator tool which assists an office in implementing its procedures”[11]. They set out to “reduce the transaction costs of communicating, coordinating and deciding” by “formalizing and automating protocols, which are plans of cooperative activity to facilitate office procedures”.

Like Officetalk it is very much a document-driven system. There are four basic concepts in the XCP system - those of person, role, actor and document. A person may perform several roles, in a similar way to PSS. Documents of pre-defined types may be created by certain roles, and they then move down pre-defined paths. Thus someone in role type A will perform certain actions relating to the document, and then pass the document either to a specific person performing role B, or back to the system to assign to someone in role type B. The system tracks the progress of the document, and keeps a history of it.

The architecture of XCP is very like that of PSS, with a separate “command interface” and a terminal-independent part, consisting of an application-independent base of protocol primitives and an application-specific protocol. XCP was implemented in VAX Lisp and runs on VAXes.

Both Officetalk and XCP are a lot less sophisticated than PSS in a number of respects. Ellis and Nutt draw a distinction between document-oriented models, and process-oriented models of office information systems. Documents are a way of organising information, and secondarily of organising activities. A document-oriented system would leave office procedures very much as they are. An automated office does not have such a great need to use “documents”, electronic or otherwise.

Neither Officetalk nor XCP address the question of change mechanisms, nor do they attempt to provide any integration facilities.

3.4 SCOOP

Zisman’s SCOOP (System for Computerization Of Office Processing)[9] is based on Petri nets augmented by production rules as a means for modelling offices as asynchronous concurrent processes. In contrast to Officetalk and XCP it is process-oriented. SCOOP is run by an ‘execution monitor’, driven by an internal representation of a set of augmented Petri nets. The firing of a transition T causes an updating of the available productions (roughly equivalent to ‘actions’ in PML). Processes may be either automatic or interactive. There is also the idea of using exterior tools such as document generators and mail services.

3 Precursors of PSS

3.1 Co-ordinator

Winograd has written several articles on CSCW the most notable of which is "A Language/Action Perspective on Cooperative Work"[10]. The main focus of his work in this area is on what he terms "conversations for action" (CfA). He has taken the idea that most ad hoc co-ordination occurs through conversations which follow certain set patterns and used a formalisation of these as the basis of a system known as "The Co-ordinator". This was designed initially to run on IBM PC compatibles. Since it is in essence a messaging system it also requires the PCs to be networked.

Winograd uses the traditional linguistic division into syntax, semantics and pragmatics in order to lay the basis for his perspective. He is primarily concerned with pragmatics. The language/action perspective emphasises that "people act through language". This has two aspects. Firstly that when people have a conversation they are doing something, often many things interleaved, such as, for instance, giving information, helping someone with their work, and establishing personal ties. Secondly, that although people obviously do not always act through language in the sense of using words, actions can often be seen as having meaning due to the social context in which they are performed, in that they fulfil obligations or set up others. They may be part of conversations.

The main lesson that we can learn from Winograd is the usefulness of formalising CfAs in order to support articulation of tasks not directly controlled by the PSS system. It might be asked why we should wish to build these into the system rather than rely on spoken conversations or ordinary email. There are two main reasons:

- Clarity. Conversations are often mixed as to subject matter, and it may be unclear (a) exactly what the request is for (b) its importance (c) its due date. Articulation often involves negotiation, and two parties may think they have reached agreement when in fact they are still some distance apart
- Keeping track of the action

Whereas a Morph-type structure provides facilities for vertical problem-definition and task allocation, a Co-ordinator-like facility provides for requests with a horizontal element, across the organisation where tasks cannot be simply assigned, but must be negotiated.

3.2 Officetalk

Ellis and Nutt give details of several automated office information systems. Officetalk was developed by Xerox PARC during 1976-77. Interestingly it grew out of a study of languages for expressing office procedures. It is based on the idea of substituting electronic forms for the paper ones commonly used in offices. Thus the user has an electronic desktop with in-tray, out-tray, a set of blank forms, and a file for forms being worked on. It would be a simple matter to develop a system such as this using PSS, and it would have wide application. Various sophistications could

information systems, as these systems become more and more like social organizations in their own right (diffuse, evolving, and decentralised, with multiple viewpoints)."

2.5 Informal interactions

2.5.1 The importance of informal interactions

By informal interactions we mean any interaction that is outside the formally laid-down description of how tasks or roles are to be performed. Obviously this is not a hard and fast demarcation, for there is a continuum from procedures that are spelt out in manuals to more or less purely social interactions. However Wynn [8] has shown that even at the most social end of this continuum the interactions which occur can be most important for the successful operation of the organization in terms of the passing of background information, and the maintenance of the organizational culture. She also demonstrates that informal communications are necessary in order to implement co-operative problem solving, and solving problems of articulation. Informal communication keeps things running smoothly, and makes it easier for people to do their jobs.

Very often, even in organizations with well defined procedures, there are informal interactions across the organisational structure that occur on a quite frequent basis for the purpose of obtaining information or facilitating tasks, and without which the organisation would run much less effectively. It may be argued that these informal interactions should be brought within the formal procedures. There will indeed be advantages to doing this in that these informal procedures may be very dependent on the particular people involved in them. However, as Ellis and Nutt[9] point out the "capturing the information content of informal conversations is neither trivial nor well understood", and thus it is not easy to identify these interactions in order to bring them into the model. The challenge for PSS designers is to build systems which incorporate the formal structure, but allow and if possible facilitate the informal.

2.5.2 The threat to informal communication

Ellis and Nutt [9] state that "automation of an office is likely to upset the informal communication mechanisms, possibly causing the office information system to fail". The primary reason for this is that work procedures may be changed in such a way that the old links are no longer relevant. Alternatively the person called on to perform a task or provide information may no longer be able to perform the task due to a lack of information or other resources, or they may no longer feel that it is in their interest to help.

As Checkland [7] puts it, “it is never possible simply to describe real-world purposeful activity once and for all”. He argues that we can talk of purposeful activity only in terms of a particular interpretation, bias, prejudice or value system, and this means that “we have to accept (a) that there will be multiple possible descriptions of any named real-world purposeful action, and (b) that any description of purposeful activity which is to be used analytically will have to be explicit concerning assumptions about the world which that description takes as given”.

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. 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 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.

2.4 The problem of articulation

“Articulation consists of all the tasks needed to co-ordinate a particular task including scheduling subtasks, recovering from errors, and assembling resources” according to Gerson and Star [6]. We need to understand the articulation occurring in any given work situation in order to bridge the gap between objectives and practice. That is to say, if our model of the activities going on in an office only includes those directly related to the achievement of objectives, then if we try to match the model to what is actually going on in the office, the fit will be very uneasy, and any attempt to build a Process Support System on such a model will be likely to fail. As Gerson and Star put it, “articulation is not extraneous to requirements analysis, but central to it”.

Articulation may be more or less formalised. Bannon and Schmidt point out that “In work environments characterised by task uncertainty, due to, e.g. an unstable or contradictory environment, task allocation and articulation cannot be planned in advance. In these environments task allocation and articulation is negotiated and renegotiated more or less continuously”.

In such situations there arises what Gerson and Star[6] refer to as the ‘due process problem’. They state that “The inherent necessity for negotiation, rather than algorithmic calculation, in order to arrive at an answer is the due process problem. The due process problem can arise at any level of organisational scale. Even relatively centralized organizational units must evolve, negotiate, and remain subject to local constraints.” They see these problems becoming increasingly common: “due process problems will become intrinsic to the organization of

organizational rationality". He goes on: "Rationality is always political. No-one is neutral in the management of organizations".This is not to say that organizations are irrational in the way they operate, but that there must always be a complex balancing-out of the interests of those with the potential to affect the running of the organization (and they may be outside as well as inside, staff as well as managers).

There is thus no simple mapping between the stated aims and objectives of the organization and the day-to-day operations it performs. This is a major problem for operations managers and system analysts and designers. A system may be devised which from a disinterested and strictly logical viewpoint appears to offer the organization great gains in efficiency, yet if the designers failed to understand the rationality behind the previous system it may prove unacceptable because it upsets the balance of interests in the organization.

Interests may coincide, conflict or be neutral with respect to one another. Where the interests of a group of people coincide, they may join together to press those interests, possibly against other groups. Morgan states that "Conflict arises whenever interests collide." However a distinction should be made between conflicts of interest (e.g. it is in the employee's interest to be paid more, and in the organization's that he be paid less), and active conflicts (such as a strike over pay). The former will always exist, but are usually accommodated by a form of balancing, in order to avoid the latter. It is this balance that designers must be careful not to disrupt.

Organizations will normally attempt to minimise active conflicts as far as possible. However, they will often make use of conflicts of interest. People are often pitted against each other in search of promotion. Groups have to compete for scarce resources. Competition is a way of promoting improved performance.

Control of technology and information are both instruments of power. Information itself is rarely neutral - it is wanted for some purpose. The giving of it serves a purpose. The development of any information system is thus likely to become a political issue in the customer organization. This is likely to be even more true with the introduction of a PSS, which may radically change work procedures and access to information. As Franz and Robey [5] point out "users may resist or sabotage systems that do not fulfil their needs, and support design features which enhance their own situations". There is thus a "need to be aware of both the rational and political aspects of the system development process".

2.3 Reconciling perspectives

Gerson and Star draw attention to the difficulty of gaining a neutral description of the processes occurring in an office. They point out that "No representation of the world is either complete or permanent. Rather, any description is a snapshot of historical processes in which differing viewpoints, local contingencies, and multiple interests have been temporarily reconciled"[6]. This, it should be noted, is the best that can be achieved, and it can only be achieved if there is recognition of the need for reconciliation of viewpoints.

to say that “the procedural structure of organizational activities is the product of the orderly work of the office, rather than the reflection of some enduring structure that stands behind that work”.An extreme position such as Suchman’s would imply that PSS is bound to fail, as it sets out to automate routine procedures. But Suchman is overstating the case. Whilst it is true that informal networks and ad hoc procedures do 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.

Morgan shows that the mechanistic image simply fails to take account of important aspects of organizations, such as the political. All employees have their own individual interests, and groups of them will have varied shared interests which they will seek to promote. As Morgan points out “Empire building, careerism, the defence of departmental interests and pet projects, and the padding of departmental budgets to create slack resources may subvert the working of the whole” if organizations are structured on mechanistic principles.

2.2.2 The organic model

In reaction to the mechanistic view recent organizational theory has tended to emphasise the organismic characteristics of organizations. One strength of this is an emphasis on the importance of the environment in which organizations operate, and that organizations are open systems, best seen as on-going interacting processes, rather than as collections of parts. Similarly objectives may be seen ultimately as processes, the primary one being survival, rather than goals which are often targets or end points to be achieved. There is also a recognition that organic, or project-oriented matrix forms of organization tend to be better at producing innovation and handling change than bureaucratic- mechanistic ones.

The main drawback with the organism as a metaphor for organizations is that organisms are functional unities - there is no conflict between parts because there is no divergence of interests. Much as it may be desired, in organizations this situation very rarely obtains. Organizations are composed of people, and people have their own interests, which will inevitably conflict to some extent with those of others, and those of the organisation as a whole. Even where there is a high degree of identification with the aims of the organization amongst its members, there is often still conflict about how those aims should be achieved.

2.2.3 Organizations as political systems

As Morgan points out, organizations are political systems. There is often a marked reluctance to admit this fact. ‘Politicking’ is generally seen as detrimental to an organization, so there is a tendency to deny that politics is engaged in at all. Even if its existence is admitted people are rarely willing to discuss the politics of their organization except in the most general terms. This is largely due to the fact that to reveal their knowledge of what is occurring, especially to an outsider, might jeopardise their own position within the organization. Also, of course, people are aware of the politics to varying degrees.

However, it is necessary to be aware of the political dimension of organizations for a number of reasons. The first of these is, as Morgan puts it, to “explode the myth of

2. PSS in the organisational context

2.1 Introduction

PSS provides a platform for the creation of co-ordinated systems. However as Bannon and Schmidt [1] point out the history of office automation systems has not been a happy one. Early attempts to automate the office failed for two main reasons: firstly because the hardware was not sufficiently sophisticated, and secondly because they adopted too simple a model of how offices, and organizations in general, operate. Hardware has now developed sufficiently for the task, but there is still the necessity for PSS system designers to find methods of building rich enough pictures of how organizations work.

PSS will be a pervasive presence in the offices in which it is installed. Whatever staff are doing will be done in the PSS context. Whereas most installed computer systems address well-defined and limited sets of functions, PSS will be involved in all, or a great deal, of what goes on. Because of this, and the fact that it is always difficult for organizations to integrate and adapt to systems of any size, a good “fit” between the organization and the PSS system designed for it is imperative. In order to achieve this an understanding of organizations must underlie the design. Thus there is a necessity to incorporate the lessons of organization theory into any design methodology to be used in building PSS systems.

2.2 Complexity of organisational functioning

2.2.1. The bureaucratic/mechanistic model of organizations

Morgan [2] presents several common images of organizations, and shows the strengths and weaknesses inherent in viewing them in these ways. The most common image is of the organization as a machine. In fact this has become a ‘natural’ or default view. Computer system designers have traditionally, to the extent to which they concentrated on function, taken a mechanistic view of the organizational processes they encountered. If this were an adequate view then there would be little problem in automating them. However, as Morgan points out, “mechanical imagery tends to underplay the human aspects of organization, and to overlook the fact that the tasks facing organizations are often much more complex, uncertain and difficult than those that can be performed by most machines.” He points out four disadvantages of a mechanistic approach; it can: “(a) create organizational forms that have great difficulty in adapting to changing circumstances; (b) result in mindless and unquestioning bureaucracy; (c) have unanticipated and undesirable consequences as the interests of those working in the organization take precedence over the goals the organization was designed to achieve; and (d) can have dehumanizing effects upon employees”.

Many authors have decried “the traditional bureaucratic model” of the office (e.g. Bannon and Schmidt [1], Shiel [3]), 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. Suchman[4] goes so far as

1.Introduction

This report sets out to review some of the literature relevant to the IOPT project in the following areas:

- CSCW
- Organisation theory
- Process Modelling
- Soft systems

There is also a discussion of the nature of process modelling, and its relation to the process of designing process support systems.

1.1 The relevance of CSCW work to IOPT

The most striking feature of PSS in terms of the benefits it can provide to users is its facility to co-ordinate the work of an organisation. In theory all this work can be brought under the direction of the PCE executing a process program. Bannon and Schmidt [1989]say that “co-operative work is constituted by work processes that are related as to content, that is processes pertaining to the production of a particular product or service ..... co-operative work relationships are characterised by being planned or rather premeditated”. Thus clearly a PSS constitutes a computer system for supporting cooperative work. 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. Usually such systems address a narrow co-ordination problem domain, such as co-authorship of documents, or support for group decision-making. PSS obviously operates in a much broader fashion than this, although a PSS system could include such facilities, either specially written or in the form of tools integrated into the system.

CONTENTS

Process Modelling and Computer Supported Cooperative Work

P. R. White

Manchester University