ABSTRACT:

In this paper we present some of the results of an empirical investigation of systems development project work. These results are the outcome of our attempts to 'listen to' a project group developing a large computer application for a bank. It has been our primary purpose to catch the informal sides of project work, and this paper will focus on two aspects: Project work as *group work* as a means of *creating something in common*, this something being a computer application which will be used in a use-setting outside the project group. A comparison of our findings with different understandings of systems development project work found in literature is included, as is a discussion of our methodology.

This paper was presented at the 11th IRIS conference in Røros, Norway, August 1988.

In computer science/systems development research the discussion of cooperation in development projects, among project group members, as well as between systems developers and users has become important. Methods and computer support for this type of cooperation plays a prominent role. We can list planning methods such as CASE tools and decision support systems[1] as examples of formal planning tools. The interest for formal ways of describing and understanding these aspects seems much more dominant than attempts to understand what happens in real life development projects where people work together. The ROSA project is an attempt to get to such an understanding.

The idea is to focus on the social and less formal sides of project work. We have tried to develop ways of helping ourselves as researchers, as well as the systems developers, focus on how people work together in a project group, how they act in real situations, what are the conditions that make people act together and belong to a group, etc. Our focus is on both the *process* of developing computer systems. The process of creating computer systems, like that of constructing buildings or any other artifact, is usually brought about by people working together in groups; groups that function within the framework of a project. Here, through the lens of the ROSA case study[2], we will look at project work.

Hopefully this can add substance to understanding systems development, as well as it has been our goal to try out and work on new types of research methods to be used in systems development research.

---

[1] See *Proceedings from the Conference on Computer-Supported Cooperative Work*, Austin, Texas 1986, and Bøgh Andersen, Peter et al.: *Research Programme on Computer Support in Cooperative Design and Communication*, University of Aarhus 1987.

[2] Bødker, Susanne and Greenbaum, Joan: *A Feeling for Systems Development Work - Design of the ROSA Project*, Daimi PB-246, University of Aarhus 1988. The members of the ROSA project are Berit Holmquist, Randi Markussen, Regine Hansen, Lisbeth Faber Rasmussen, and Tage Stephansen, and the authors. The ROSA project was born out of the research program on computer support for cooperative design and communication at the departments of Information and Media Science, and Computer Science at Aarhus University.

A ROSA Perspective

The ROSA project was set up as a small pilot study to look at the informal working practices of systems developers. The name ROSA, a Danish acronym standing for roles and cooperation, reflects our interest in the social dimension of systems work. Our interest is on getting a closer look at how people work together in developing projects, how they act in different situations within the pressure-cooker of project development, and how they share or distribute knowledge.

This shift in emphasis, from the traditional focus on formal methods to a spotlight on informal project activity, gave us a chance to try to understand the nature of project work as a connected network of situation-dependant activities. Our approach was inspired by Evelyn Fox-Keller. Keller, in describing the research of Barbara McClintock tells about research methods for 'feeling for', or 'listening to', the subject of the research.[3] Barbara McClintock, a well-known, although unconventional scientist, marked her research with a strong belief that one should be 'letting the material speak to you'. Instead of descending on a research topic with a set of preconceived categories, McClintock choose to think of herself as being part of the material. For her, scientific work was living and being involved in the subject matter, or what she referred to as 'getting a feeling for the organism'.

We have tried to work in a similar way. We started from our personal involvement in our work and the interests of the participants and used this involvement to frame the questions, issues and hunches that we felt were important in the social construction of computer systems. Our research approach, like others, obviously begins with our subjective feelings about the topic. Yet we have attempted to follow McClintock's footsteps in trying not to overlay preconceived categories on top of the existing social complexity. Our interest is not to build a model of the complex project world, but rather to try to understand it. As our point of departure we have looked at the

---

[3] Fox-Keller, Evelyn: *A Feeling for the Organism: The Life and Work of Barbara McClintock*, San Francisco, W. H. Freeman, 1983.

working relationships for both men and women system developers. In doing this we wanted to focus on four aspects of informal working relations that often are associated with women. These four: intuition (which, for us, is experience-based use of practical understanding), learning processes, language use and roles (ways of acting in different situations), formed the themes of our research project. Using these themes as a working definition of informal work relations allowed us to highlight aspects often kept out of the research spotlight.

Our research approach, with its emphasis on 'feeling for' the material, we call a gender perspective. The use of the word *gender* implies that we are not only interested in women, per se, but rather in the way that social reality is framed by cultural myths about men and women. Since we have written about these myths in other articles[4], here we would rather highlight some of the differences between our use of a gender approach and the approach taken in 'traditional' computer science and systems development research. In doing so we are making our assumptions clear, while trying to avoid the trap of pre-framing our subject.

Clearly our perspective is a hermeneutic perspective by being 'interpretation based on pre-understanding'.[5] Methodologically it resembles other hermeneutic traditions, emphasizing in particular the aspect that you, as a researcher are involved and acting in the situation.

The following illustrates some of the differences between our approach and the commonly followed path of computer science/systems development problem solving, such as the systems approach of Yourdon, for example. As Yourdon puts it, top-down design, is "a *design strategy that breaks large complex problems into smaller less complex problems and then decomposes each of these smaller problems into even smaller problems, until the original*

---

[4] Bødker and Greenbaum, op.cit., Greenbaum, Joan: *The Head and The Heart*, DAIMI PB-237, University of Aarhus 1987, and see also Fox Keller, Evelyn: *Reflections on Gender and Science,* Yale University Press 1985.

[5] Winograd, Terry and C. F. Flores: *Understanding Computers and Cognition: A New Foundation for Design*, Ablex Publishing Comp. 1986.

problem has been expressed as some combination of many small solvable problems."[6] These approaches, furthermore, assume that complexity in systems requires *scientific study,* separate from daily life. The ROSA approach sees complexity in systems similar to complexity in life, i.e. computer systems are not normally conceived as complex in their use situation, and a computer system as part of a *social, political and organizational setting.* Therefore, systems development, as well as systems development research must focus on *people in situations.*

We are departing from some newer, less structured approaches to systems development, such as those described in *Professional Systems Development,* which while viewing systems projects as part of a larger social and political world, focus their attention on a set of prescriptive methods that developers can follow when planning a project.[7] At the same time we are joining the choir of researchers, for instance T. Winograd & F. Flores[8], L. Suchman[9], P. Ehn[10] and H. & S. Dreyfus[11] who consider human activity, including design or use of computer applications, not as primarily characterized by rationality, planning, and reflection, but by practice and our ability to act in situations, which are more or less familiar to us.

What we have been particularly concerned about is our pre-understanding, and how that affects our ways of acting in the research situation. Our pre-understanding as women researchers in systems development is that there are less-prestigious, less-formalizable, social activities going on in systems development, activities that we know very little about. Our inspiration from

---

[6] Yourdon, Edward: *Managing the Structured Techniques*, Yourdon Press 1986, p. 61.

[7] Andersen, Niels-Erik et al.: *Professionel Systemudvikling*, Teknisk Forlag 1986 (Professional Systems Development).

[8] Winograd and Flores, op.cit.

[9] Suchman, Lucy: *Plans and Situated Actions: The problem of human-machine communication*, Xerox ISL-6, 1985.

[10] Ehn, Pelle: *Work Oriented Design of Computer Artifacts*, Swedish Center for Working Life 1988.

[11] Dreyfus, Hubert L. and Dreyfus, Steward D.: *Mind over Machine–the power of human intuition and expertise in the era of the computer*, Basil Blackwell 1986.

Fox-Keller and McClintock tells us that we should not stand too strongly on this pre-understanding, but enter the process with an open mind. We cannot see all our prejudice ourselves, but we think that having been brought up as women and not as men has helped us in the process, posing different research questions than those most often put by men.

In the following, we turn our attention to a short description of the case study and the nature of project work, then we return to reflections again.

The Case Study – A Bank System

In the ROSA case we investigated a systems development project in a large bank. The project, which we refer to as the Transaction project, is a one year effort. The systems development work is being carried out by approximately 30 systems developers, organized in smaller groups with different, but overlapping, tasks, and managed by group leaders. The project group grew very fast within the year, starting out as only a handful of people. The goal of the Transaction project is to come up with a system to replace a 10 year old application which provides terminal-based customer record-keeping throughout the bank's more than 200 branches. Within the bank, the Transaction project is part of a Systems Development department which employs 120 people.

The Transaction project is formally split up in two big development groups and a smaller Test and Users group. The development groups: Decentralized and Centralized systems are separated according to technical tasks. The decentralized group deals with programming of personal computers which are to act as front-ends for the big centralized system developed by Centralized systems. Contact with users is maintained primarily through a group handling bank practices in general, and through a reference group, consisting of experienced users (called, 'super users') selected from the bank branches.

The department's management is interested in improving its working methods, and a new general method and documentation standard is currently being introduced into the department. Time plans play a key role in the management of projects. Each individual project member has to plan and schedule his or her work tasks into pieces no longer than ten days.

The systems developers are trained either as business assistants with additional systems development training, or as programmers, engineers, or computer scientists. They equally participate in all aspects of the development and implementation process, with little division of work between systems analysis, design and programming; generally working in small groups that cut across the the formal project organization lines.

Our ROSA research project was set up as a small pilot study. We lived under the conditions given by the case. One of these conditions was that there was very limited contact between the systems developers and their prospective users. We discuss some problems with this later. Here we shall just note that we had no possibilities of studying the cooperation between users and systems developers.

The research spectrum of ROSA is much broader than what we talk about in this paper. We have chosen a range of research techniques that bring together a group of inter-disciplinary researchers in order to get a variety of qualitative interpretations. The project applied conversational analysis from linguistics, qualitative interviewing techniques from the humanities, and action-based workshop methods from the social sciences. *Conversation analysis and observation* of different meeting situations were used in order to focus on language and roles. The in-depth *interviews* with systems developers were applied to focus on their work life, home life, and qualifications.

The use of *workshops* and participant-observation were intended to let the participants set their own agenda. Techniques like role-playing, diaries and story-telling, and future workshops were used to help the systems developers raise such issues.

In this article we shall focus on the work in these workshops. The group consisted of 8 systems developers, 4 men and 4 women. In most of the workshops the number of participants were 5 - 6 with each workshop lasting for two hours. The systems developers came from different educational backgrounds, but none of them were formally trained in banking.

In some ways we can see the sequence of workshops as one long future workshop[12], spanning the phases of critique, imagination and realization.

We started out with first part of a future workshop which is a brainstorm activity where the participants write keywords on big sheets of paper visible to everybody. After that the participants structured the keywords together and gave them headlines. The idea is to catch spontaneous reactions. The first brainstorm was: what do we like about our work? The second brainstorm was: what don't we like?

In the next workshop we used a picture game to deal with description and interpretation. We then moved through a workshop on organizational culture, where story-telling was used and where the participants made their own culture analysis of their organization. The next workshop was on women and men as systems developers, where role-playing was applied. Using the maps of the MARS project[13], the participants in two groups described some of the problem situations that they have encountered in the project.

The main problem seemed to be that of communication up and down in the hierarchy, and that of the time plans which didn't hold. Frustration, confusion and too little time are important keywords. New people were, in particular, frustrated, and didn't get much introduction. The lack of understanding for the product (the system) became evident in the workshops. Some work is done in the project to help people understand how the things that different people do relate to each other, but not for an understanding of the whole of the product and its use.

Although the intent of the ROSA project was to study the role of intuition, learning, language use and roles, our focus shifted to an emphasis on co-ordination and communication, as these were the issues important to the

---

[12] Future workshops are developed by Jungk, Robert and Norbert R. Müllert: *Zukunftwerkstätten, Wege sur Wiederbelebung der Demokratie*, 1981. Their idea was to help community groups of different kinds to understand their own situation better, and to change this. The main technique in future workshops is structured brainstorming focussing on critique of the present situation, visions about the future, and realization of these visions.

[13] Andersen et. al., op. cit.

systems developers in the bank. Of the original themes, language use and roles were dealt with in the workshops, whereas learning came to be of no importance, except in the narrow sense of getting information on project status and plans. Intuition seems to be a too fuzzy concept for what we encountered in the project: rather it seems fruitful to talk about experience-based action in situations.

Reflections

In the following we elaborate further on some of these points. These reflections are based on the interpretation that we, together with the group, did on the so-called communication problem, and what it means to be part of an expanding project group which is producing a computer system for somebody else to use.

The observations that we have chosen to discuss are heavily intertwined, but, as writing is a linear process we have to start somewhere.

The group doesn't know the product they are creating

Typically, only the very few people who took part in the start of the project knew something about how the new system is intended to function in the bank. The rest of the group had little idea of how their product would change the daily work in the bank. Their knowledge of the product was purely technical and mainly they knew only their own modules, and which (other people's) modules they were supposed to interface with. One of the things that we noticed in the Transaction project, was that the systems developers were lacking an overview the system as a whole. Some developers thought that their problem was one of communication and coordination. If they could just find out more about what the other developers were doing they would find out how the parts fit together.

This can be interpreted easily as pure alienation of the systems developers, and it is. As with much work it is assumed that the system developer can do her task without gaining much familiarity with the product. However, the product, the computer system will be used again and again in situations which are, in this case, not known to the systems developer, and, in general,

not fully predictable.[14] In other words it is a necessary part of the skill development of a professional systems developer to gain experiences with the product and use this in following projects, but it is also important for the systems developers as individuals and as a group to know the product that they are working on, because a mutual understanding of the product is necessary for keeping the group together. For systems developers working on a corner of a computer application, that corner, or function, takes its usefulness from the overall system. As a systems developer it might be similarly confusing to know about the technical subfunctions of a program without knowing what that program is expected to do within the overall system. Two aspects of this needs further investigation: Systems developers see themselves not as industrial workers, but as professionals – how can they develop their skills as they move along? Systems development is a process of moving from one project to the next - how do systems developers see the product they are creating?

Literature on how to plan and construct computer systems, dwells primarily on tools and techniques for separating the technical parts.[15] This body of literature, aimed generally at the practitioner or managerial level, focuses on 'how to do it'. While it's not followed to the 'letter of the law', it is quite influential in directing management policies about how systems projects ought to be carried out.[16] As we saw in the Transaction project, for example, the emphasis on computerized time reporting and a formal system for creating program modules, reflect this practice.

---

[14] Bødker, Susanne: *Through the Interface – a Human Activity Approach to User Interface Design*, DAIMI PB-224, University of Aarhus, 1987.

[15] See for example, Yourdon, op.cit., Jackson, Michael: *System Development*, Prentice-Hall 1983 or Lundeberg, Mats et al.: *Systemering*, Studentlitteratur, Lund 1978 (Systemeering).

[16] See, for example, Greenbaum 1979, op. cit. and Friedman, Andy and Greenbaum, Joan: *Datamation* series on ICON project. (*Datamation*, September 1984, Vol 30, No. 14 and 15) and see *ICON Working Papers*, Bristol University, Economics Department, Bristol, England.

General systems theory, on the other hand, attempts to lay the theoretical groundwork for the way parts or components should be fit back together.[17] But in doing this, they tend to concentrate on the formal, structured ways that parts become a whole.

Newer systems development literature talks about the need for users to understand the system even before it is fully developed[18]. Prototyping and fourth generation tools are discussed as ways that users can interact with a developing system and experience their future work situation. *Yet, it's not just the 'users' who need to know about the system, but the systems developers themselves.* We believe that their problem is actually quite common in project work. Communication and coordination problems develop *because* work is focused on parts of the system which must be put back together through coordination. Some relatively simple suggestions like using mockups and prototypes[19], situation plays and scenarios might give members of the project team more of a chance to focus on the overview, rather than their own, more narrow, part.

[17] See for example, Checkland, Peter: *Systems Thinking, Systems Practice*, John Wiley and Sons 1981, Churchman, C. West: *The Systems Approach*, Dell Publishing Co. 1968.

[18] Ibid.

[19] See e.g Kyng, Morten: *Designing for a dollar a day*, presented at the CSCW '88, Portland Oregon.

Waiting for the baby to be born (a feeling of belonging)

The development of a project, often called its life-cycle, is usually described in the literature as a series of phases leading up to the final testing of the system. But to anyone living inside of a project it often feels as if project life where more like riding a roller coaster. There are periods of relative slowness, and in fact boredom, when the details of specifications need to be worked out, followed by intense periods of high-speed activity where parts of the system must come together and roll through the hoops of testing. The deadline is the powerful motor of project as it rolls and rushes toward becoming a workable and usable computer system. And it is not until the end, the point where one can see and use the actual product, that involvement or a sense of belonging usually happens!

When we met with the project members in the Transaction project they were three-quarters of the way toward the deadline, just about to begin the final roller coaster ride toward testing. Up until that point the project had rolled by with periods of ongoing small meetings where developers discussed details of program and project coordination. Perhaps the bread and butter of project work, but certainly nothing exciting. People commented about their close colleagues from former projects and about how they would stop work on this project when a problem popped up on an earlier system. They mentioned stories about projects that were completed and friends and colleagues that they worked with before. In short, without the excitement of creating something in common, the sense of belonging to the project is slow in forming.

We don't think that this is unusual for the Transaction project. Rather we feel that the nature of project work, with its stark time contrasts, its deadlines, and the commitment to transforming a vision into a concrete common product, must be fully realized in discussion of project life-cycles. The social dimension stands out quite boldly when we realize that the *feeling of belonging* to a group is associated with the interaction of actually creating or doing something together. And that this doesn't happen, or might not be acknowledged, until near the end of a project's life. For systems developers, whether one works for a large organization or a small consulting firm, the

end of a project means the beginning of the next one. And so the problem of belonging simply rolls along. Perhaps if we compare this with a woman having a baby and giving it up for adoption, we can see the abruptness of project life.

In a traditional systems life-cycle approach once a system has passed testing it is 'turned over' to a maintenance team for handling the occasional and necessary technical repairs. Maintenance work is, or has been, viewed as something not as technically sophisticated or demanding as development work. But in changing our view of project life-cycles we can see that what happens to a system once it passes testing is like watching a baby grow. Systems change in use. Users change as they interact with the system. Organizations adapt to the introduction of computer systems. All of these activities are critical to our experiences as designers of computer systems, because if we don't know, we may easy repeat the same mistakes. By considering the post-testing phase less important we sweep into the background the social use of computer systems, and the social process of creating something in common.

An expanding project

In the Transaction project traditional administrative procedures define, and allocate tasks to the individual. The time sheet and work-load reports that each person must fill in on a weekly basis, effectively place responsibility on the individual level. Thus posing a stark contrast between the collective responsibility of the group in getting the system 'up and running', and the pressure on the individual to be responsible for only his or her own part.

Management literature in the tradition where this bank's management belongs, treats project leadership as the act of monitoring individuals, although it is obvious that the product, the system itself, is a result of a group process. Coping with an expanding project in this perspective is a matter of dividing up tasks so that a new participant (as well as anybody else) can conduct her tasks knowing as little as possible about the history of the project, the product and other participants' tasks.

Most management time sheets and formal reporting mechanisms rarely record what actually happens on the job.[20] Within the Transaction project, for example, probably the largest category of office jokes cluster around the how little the real work resembled the 'official' time records.

Another, more subtle level of group work remains as both a problem and a possibility of group activities. Through the workshops and the in depth interviews in the ROSA project we repeatedly found the issue of 'new group members' popping up. The general theme, expressed in a variety of ways, tended to cluster around the feeling that project members felt that the project was growing so fast 'they didn't belong to it'. Some said that 'there were so many people they didn't know in the project', and others complained that 'they were closer to people that they worked with before, than the people on this project'.[21]

After analyzing these comments, and discussing it with the project members, we realized that while we were looking at the traditional problems of a project that grows from a few to 30 people within one year, we were also looking at a fairly common problem of *belonging*. Belonging to a group, whether it is a family, community, or set of work colleagues, is an important element in the normal functioning of any human activity. Belonging to a group means to have achieved things together; to know the history of the group. This both explain why people don't feel that they belong to the group until the end, and why new people need to know more than just the task they are to conduct. They need to know the product they are part of creating and the history of the process they are part of.

In traditional methods, cooperation between project members is dealt with much in the same way as management in our case study has done it, through:

---

[20] See, for example, Greenbaum, Joan: *In the Name of Efficiency*, Temple University Press, 1979.

[21] Rasmussen, Lisbeth and Stephansen, Tage: *Den sociale dimension i systemudvikling. - Et casestudie om grupper og kultur i en udviklingsorganisation*, DAIMI IR-79, University of Aarhus 1988 (The Social Dimension in Systems Development).

• specification methods, decomposition into parts of complex systems. These parts may be called entities/actions[22], information sets/precedence relations[23], or data flows/data processes[24],

• phases, time structuring of projects,

• division of work, structuring of work as to include different people in different phases of a project (e.g. chief programmers' teams),

It has been known for years that the application of such approaches causes immense problems, and that the gap between such recommendations and the real world is very big. From ROSA we have seen that it is the start up, and closing down of a project, which causes the biggest problems. Almost all methods deal with running projects.

Introduction of new people is dealt with only as a problem of avoiding spending time on them (how to define limited tasks that somebody can conduct without knowing the rest of the project). The need for systems developers to know more about what they are doing than just the tasks that they are to undertake, does not seem to be recognized by even the most 'humane' methods.

The MARS project[25] did empirical investigations of several projects at 4 Danish systems development firms. The project was a much bigger effort than ROSA and the aims were to develop an understanding of, and recommendations for how professional systems development should take place.

The main problems found in the empirical work with systems developers in the four firms deal with

• project start is often too fuzzy.

---

[22] Jackson, op.cit.

[23] Lundeberg et. al., op. cit.

[24] Yourdon, op.cit.

[25] Andersen et.al., op.cit.

• there is a need for a continuous planning and evaluation process as part of the systems development project - a time and resource plan can not be made once and for all.

Taking a first look at the MARS project, most of the recommendations given by the MARS literature would be useful for our case. Furthermore, the research methods resemble ours, such as working groups with systems developers. Their effort was of course a much bigger one than ours including experiments with actual change of work methods in some selected projects. Yet, the research results and recommendations that the project came up with was very much on the formal and rationalistic side[26]: recommendations for structured reviews, and other ways of estimating and structuring systems development projects rather independent of the actual situations. From a ROSA perspective, listening to systems development work means that the participants need to know the history of the project and a fuller understanding of the products.

Cooperation in project groups

We have been circulating around issues of when systems developers feel that they belong to a group and achieve something in common. Clearly, it is our assumption that the cooperation and social interaction among systems developers is an important part of their daily work life. We have seen this confirmed in this case, where people have emphasized the 'not belonging' of new and old group members, and their identification with former project groups.

Engeström[27] in his book *Learning by Expanding*, identifies three forms of intersubjectivity, or being together, in work groups: 1. *Coordination*. Individuals are together to act upon a common object, but their individual actions are only externally related to each other. They still act as separate individuals, each according to their individual tasks. Interaction occurs

---

[26] Rationalistic thinking is closely related to the masculine, as discussed by Fox-Keller, 1983, op.cit., and by Bødker and Greenbaum, op.cit.

[27] Engeström, Yrjö: *Learning by Expanding*, Orienta-Konsultit, Helsinki 1987.

mainly in the form of spontaneous reactions and attachments. For example, two programmers who share systems documentation, must coordinate when they need to look in the documentation, but otherwise they can work independently on their own task.

2. *Cooperation*. All individuals relate to an over-individual task in doing their individual tasks. The individuals have to balance both actions and the outcome of their partners actions with their own. They must influence actions and results of their partner if necessary, again with regard to the common task. There are conscious, goal-directed sequences of interaction, aiming at successful joint completion of given tasks or successful joint solution of given problems. A group of programmers work together on pieces of a program to create a certain product. They have to interface to each others programs, decide how they use variables, how the product is to appear and function, etc. A problem in one part of the program, can be solved by help from others, and may also influence the work they are doing.

3. *Reflective communication*. The living knowledge of people here develops in spoken and other symbolic processes. It becomes concrete as "we-in-the-world" or collective subjectivity. The laws of functioning of the "we-in-the-world" are not so much through inner structures of the individual's consciousness as through external practical activity involving objects and through collective cognitive activity with a shared practice. An example here could be a research group who are developing and using a programming environment. They have their own history and language, meeting patterns, and 'who to ask' patterns, and they work together gaining experiences with the programming environment at the same time as they are building it.

In the ROSA case, the project group belongs at most to the level of coordination. There is little effort made, by management, to have them belong to the level of cooperation: they don't know the product, and formally they are not supposed to help each other or coordinate with each other, once the initial specification is made. We cannot assume of a project group that they have a feeling of belonging together that doesn't have anything to do with what they are doing together. A project group is rather randomly brought together and does not have an a priori shared understanding of the

material/product and the interaction among group members. The understanding of the relation between the parts of the project that the individual members are doing, and of the product as such, are among the conditions for moving to the levels above. A shared understanding of the history of the project group is a prerequisite for the level of reflective communication.

The way we can interpret the present interaction in the project group is that it is primarily a relation within groups which exist already, not having anything to do with the present group. This means that it is hard for newcomers to be integrated, and also that most people don't feel that they belong within the (project) group.

Shared knowledge

We can interpret the results of the ROSA project, in the concepts of *situated action* rather than planned action, and *shared knowledge* instead of a focus on authoritative knowledge. These concepts arise from research done by Suchman and Jordan about such diverse activities as women's ways of knowing during childbirth and the ways that office workers communication knowledge to each other.[28] The concept of situated action and situated knowledge takes its starting point in the fact that most human activities don't happen according to preconceived plans. When talking on the telephone, for example, one may begin with an idea about what will be said, but obviously the content of the conversation will change based on what the other person has to say. As Suchman and Jordan point out, activities based on changes in situations are characteristic of most office work, and, indeed, highlight its essentially social nature. They argue that: *"Current computer systems are based not in an understanding of the practical, situated accomplishment of office work, but in a priori, rationalized models of the office as a system of disembodied information flow."* [29]

---

[28] Suchman and Jordan, op.cit.

[29] Ibid.

18

The Dreyfus brothers, in *Mind over Machine*, make a similar, but somehow more general point when they tell us that experts always act based on the situation. Novices and less proficient persons generally need to make plans that they follow, but experts act primarily on previous experiences with similar situations.[30] The relation between planning and situated action in systems development work can be seen in the following: While projects begin with careful planning about deadlines and objectives, the participants must function within the day-to-day chaos of change. Participants in the Transaction project repeated the most frequently heard complaint of systems developers – that specifications, deadlines and plans were always changing. Frequent meetings attempted to keep project members up-to-date with technical and planning changes, but meetings and their minutes don't always reflect the full range of activities that get changed.[31] In other words systems development methods and management practice deal with systems developers as though they were novices, which they are not although they often find themselves in unfamiliar situations where planning is needed.

The emphasis on shared knowledge, or the social distribution of knowledge instead of authoritative knowledge presents another contrast with formal systems thinking. Suchman and Jordan define authoritative knowledge as "the knowledge that is considered legitimate, consequential, official, worthy of discussion and useful for justifying actions by people engaged in accomplishing a certain task of objective". Authoritative knowledge in the Transaction project is for example time planning issues, and division of work. Shared knowledge concerns issues like the successes and failures of previous projects.

In general, we see authoritative knowledge as being the main shaping mechanism behind systems development management literature and a driving force behind systems theory. In looking at authoritative knowledge, we can see systems development methods' emphasis on hierarchy within systems and

---

[30] Dreyfus and Dreyfus, op. cit.

[31] Hansen, Regine: *Communication as a Precondition for Joint Action,* Department of Computer Science, Aarhus University, forthcoming.

the need for communication and control, which are illustrative of the ways that military organizations and bureaucracies place authority at the top of the pyramid. Yourdon's writings about 'top-down' structured techniques bring these principles to the project level. This need for dominance or control over the projects can be traced, according to Fox-Keller, to traditional male, scientific thinking.[32] In *Reflections on Gender and Science* she wants to show that the history of modern scientific thought takes its origin in the period when science drew away from the so-called 'irrationality' of women's knowledge and towards control over nature through reason and scientific procedures. Thus was born the age of science which replaced a social distribution of knowledge, with authoritative knowledge as represented by control over nature, people and things. And systems thinking grew as the child of this scientific rationality[33].

In systems development, scientific rationality and authoritative knowledge exist, and are important, in particular because it is through these that management conducts its job. Shared knowledge can be seen as bound to other forms of rationality which are also indeed part of belonging to a group: caring relations among group members, or between group members and users. In her book, *Caring. A Feminine Approach To Ethics & Moral Education*[34], Nel Noddings discusses caring as a philosophical alternative to the scientific rationality. A person cares about somebody by taking on this person's situation, based on her former experiences with caring. Basically we can only care about somebody, when/because we, ourselves, have been cared for at some point of time in our lives. The caring relation is, in other words, not a relation between equals, and we cannot just decide to care about each other as an explicit or implicit commitment. Acknowledging the care rationality in a project group could be one way by which we can see the use

---

[32] Fox-Keller, 1985, op. cit.

[33] See also Ehn, op. cit.

[34] Noddings, Nel: *Caring. A Feminine Approach To Ethics & Moral Education*, University of California Press, 1984.

of shared knowledge, but also one way of seeing when a group has a feeling of belonging together.

Workshops as a project tool

We agree with the focus of the MARS project on developing new methods for project organization, however we do not feel that more formal procedures are solutions. We find that active ways of strengthening the ties between the project participants is a more valuable way of creating a better work situation for the participants. This can be achieved through *knowledge about the product and the project history*.

To proceed from here we find it important to focus more closely on the creative aspects of systems development; getting closer to the content of systems development, and involving the relation between systems developers and future users. For this purpose a further development of and use of creative research methods is needed.

We found that the workshops we conducted in the ROSA project were very helpful in understanding (or 'listening to') a number of problems in systems development[35]. It was clear that there was a great need among the systems developers for discussing their problems. We have a strong feeling that any brainstorming activity would be helpful because it is important to catch spontaneous reactions. Furthermore a variety of different brainstorming approaches serves to bring about different problems, and also open up different perspectives on understanding problems.

We find that it is an interesting prospect for systems developers to include such workshops in their own practice, when it comes to the start-up of projects, the evaluation of ending projects, and to creativity and work coordination in the process. Such a use of workshops, including brainstorming, storytelling, and role-plays would provide project members with the possibilities of handling both organizational issues, such as how to work together and how to split up work, and issues about the system created. Clearly such methods must be adjusted to each project and situation.

---

[35] A report on these aspects of the project is forthcoming.

Using a gender perspective

Using the ROSA perspective described earlier we tried to show that the construction of computer systems can be seen as in a more situation-based and interconnected way. We focused on the social construction of computer systems by looking at the process as one where *group work* plays a key role, and where the end result is one done by *creating something in common.* In doing this our approach centred on *people*, the *situations* they found themselves in, and the ways that they *shared* knowledge. Since the ROSA project was just a small pilot study, we cannot offer many concrete examples of the ways these activities occur. But we think that we have ample evidence for saying that our approach is more than seeing the world through 'rose colored glasses'. For we believe, that many systems developers, like those at the bank that we studied would agree, that the systems development process can be enhanced by focusing more on an *overview* of the system being created, and, in addition, in paying closer attention to *group-based activities* rather than management's view of bite-sized and controllable tasks for individuals.

Our aim has been to understand systems development better and to help systems developers cope better under the conditions of a traditional management style. It has *not* been our aim to ask for more female values to be put into management strategies, yet the reflections point us to values that fall under the scope of interest often attributed to women. We believe that instead of putting more women in management positions, or teaching more managers female values, we need to recognize the group based activities of systems development work and develop supportive activities that enhance its potential for shared knowledge.