RESEARCH METHODS IN INFORMATION SYSTEMS
E. Mumford et al. (Editors)

A2 TWO RESEARCH METHODOLOGIES FOR STUDYING USER DEVELOPMENT OF DATA SYSTEMS[1]

Per Flensburg
University of Lund
Sweden

Abstract

Office work in organizations are guided by both explicit rules and tacit rules. When developing data systems in the traditional way (i.e., using the systems life cycle model), it is not possible to take tacit rules into account. It is argued that if this is to be done the users must be involved at a very high degree. User development is the highest possible degree of involvement. In order to know and understand the effects of user development, the paper proposes two approaches, the first one being a collection of experiences in the case study form, the second one being active participation in a user development project. The methodologies used are described and some preliminary results are reported.

---

[1]This research work has been supported by The Swedish Work Environment Found and The National Swedish Board for Technical Development.

This report was submitted to the colloquium "IS Research—A Doubtful Science" arranged by IFIP WG 8.2 in Manchester, 1-3 September 1984.

Theoretical Background

Work and Tacit Knowledge

Stig Lindholm (1979), a Swedish philosopher of science, suggests when starting research that the following questions be asked: What type of phenomenon is this? What is it a special case of? To this I have added a question of my own: What does the special case consist of?

In applying the questions to data systems, I have found two common answers. Firstly, a data system is a special case of communication. This approach is taken by among others Langefors (1966), Nissen (1976, 1984), Goldkuhl and Lyytinen (1983), and Sandström (1984). It is a special case of communication, due to the fact that the data system transforms the content of the message and stores the messages over time.

The second answer is that a data system is a special case of work description. This approach is taken by Nurminen (1981), Göranzon (1983), Cooley (1980), and myself. It is a special case of work description because it is the complete formalization of it. The two answers are, however, complementary, not contradictory! This must be heavily emphasized!

Thus I consider a data system as a complete and deterministic description of certain work and the work will be performed when the system is working, i.e., when the programs are executed. A prerequisite for work to be described in programs is a totally predictive and deterministic work. The type of work that can be described this way are all types of mechanical work operations such as mass transactions, routine work and standardized operations.

Many of the shortcomings of data systems use are due to discrepancies between description of work and intended performance. In order to make a feasible data system, it must be decided if the actual work *is* possible to describe in a programmable way. Methods or techniques to achieve this are not available today.

What types of work can be described in such a way? The work of a conveyer belt surely can. But can for instance invoice work? Some manual work might be described in this deterministic way, but I do not think very much office work can. This is due to the possibility of different interpretations of data. People performing office work are often themselves unaware of this different interpretation. Let me give an example.

As a systems consultant in an enterprise, I helped develop an invoice system. It worked with three files: a parts file, an invoice file and a customer file. Every customer was given a discount of a certain percentage on the total sum of every invoice. The people in the enterprise assured me that this percentage was fixed for every customer. It was decided, therefore, to supply this automatically by storing the percentage in the customer file. However, when the users saw the actual invoice they said: "This is wrong. The customer shall not have x% of discount, he should have y% in this case!" So the rules, which were supposed to be absolute with no exceptions, were, when applied, found to be incorrect. In fact the system had to be changed so that the actual discount was specified by the user for every invoice and not performed automatically by the system.

The mysterious thing being that the users said the rules were absolute and obviously thought this was true. But when the rules were applied unconsciously, they realized the impossibility of the results. The given rules were overridden by other rules, which the

people only realized when they were violated. The knowledge gained from this observation was the identification of at least two different kinds of knowledge:

1. Tacit rules: these are rules that we are unconscious of until they are broken (cf "tacit knowledge" by Polanyi 1966). The same phenomenon occurs when we speak and say something that is not right. The rules are mostly not possible to formulate in other ways than "It is not correct to say so and so" or "You could say it this way or that way or maybe that way too."
2. Explicit rules: these are rules that can be formulated unambigously and comprehensively.

Data systems can only take explicit rules into account. Thus data systems can *correctly* deal *only* with job steps that are guided by explicit rules. Work at a conveyor belt can be described that way, but not office work. Office work is supporting human beings and since they are partly unpredictable, this work must have a certain degree of adaptation, that is, it has tacit rules, these rules being manifest as "management style," "organizational behavior," etc.

It seems as if this fairly obvious fact is often neglected in current research on information systems. Much research is devoted to the problem of constructing a technically efficient data system based on a given requirement specification. Also a lot of research is devoted to the problem of how to make the users of the system accept it and not complain too much about its shortcomings, but instead be happy about its merits. In fact almost every data system has great merit in removing a lot of routine work. However, often other work that is yet more routine is introduced. But the deep philosophical problem of what will happen when knowledge of a certain type is unconsciously replaced with knowledge of another type or simply removed is very seldom discussed.

The Role of the Users

Tacit rules not taken into account when a data system is constructed will cause "tacit misfits." The only people that can detect and correct these misfits are the users. The correction often requires considerable changes in the data system. Due to the heavy backlog of most data departments, it often takes over a year before the changes are implemented (Martin 1982). This prolongs poor job performance; it causes a lot of problems for the users and the organization in adapting to a changeable environment. The cost of this inflexibility is often very high. The possibility for making rapid changes is thus very important. The capacity for changing data systems must be increased. This can be done in three ways:

1. By increasing the productivity of the system designers.
2. By increasing the number of system designers.
3. By both increasing the productivity and number of system designers.

The first one is possible since we today have powerful tools for speeding up the development. We have different types of structured design that might increase productivity by 10% to 20% (Martin 1982). We also now have application generators or fourth generation language that will typically reduce development time 50% to 80% (Martin 1982). Even non-data experts can use these tools, after a short introduction, to develop their data systems and thus achieve the third possibility. The conclusions so far are:

1. Office work is guided by both tacit and explicit rules.
2. Tacit rules cannot be represented in any type of computerized information system.
3. The failure to take tacit rules into account means that work will not be done in accordance with the goals of the organization and it will be difficult to change the data system.
4. In order to take tacit rules into account in a data system, the data system must be easy to change.
5. This flexibility can be achieved by using an application generator and letting the users themselves do most of the changes and perhaps the development too.

What must be shown is the common existence of tacit rules and their importance in the work of an organization. This cannot be shown by traditional positivistic scientific methods, since they only take explicit measurable phenomenon into account. This problem is discussed, among others, by Nissen (1984). We thus face the dilemma of trying to see the invisible, measure the immeasurable and describe the indescribable. Possibly the work of Tarski or Gödel in identifying the shortcomings of the empirical positivism would help us at this point. However, such philosophical discussions are beyond the scope of this paper.

A common objective of a data system is to ensure that certain work is done according to the formulated explicit rules. This is achieved simply by letting the computer do the work. A systems analyst described it this way: "Why give the human a chance of doing a job wrong, when a computer can do it right?" I call this type of data system a control data systems. In such a system, the humans in the organization have to serve the computer with data in a certain format in order to make the computer do the work. The users are servants of the machine. There are, however, systems where this is not the case. I call these support systems. The purpose of such a system is to help the user to do his or her job. Thus the user is the complete master of the system.

The functions of a support system are many. During empirical investigation, I have found the following:

1. To extract reports from common, or at least partly common, databases (query languages, report generators).
2. To produce statistical diagrams according to extracted or computed data (graphical packages).
3. To compute new figures from old ones according to an understood model of the organization (spread sheet, decision generators).
4. To retrieve information from other data banks or from an individual's own data collection.

The common database has a strength, but also a weakness. A strength, since it is possible for people to communicate and exchange information, a weakness since the interpretation of the data representing the information is determined in advance by designers of the database. It is thus a control data system. The common database might act as a gate keeper, but it may be unable to provide the desired information if it has not been anticipated by the database designers. A support system is a highly individual system. Using a common database perhaps violates the possibilities for the individual person to treat the data according to his or her own apprehension of reality. A common database implies a data model with certain information objects, properties of these objects

and dependencies between them. This model is mostly build by systems analysts and not user of the data. Important tacit knowledge might disappear in this process. Alternative ways of building such models are suggested by some researchers (Lyytinen and Goldkuhl 1983).

A management system for a common database usually presupposes integration of *all* data in the organization. Individual data or data used by just a few people do not need to be accessible by all other people in the company. The risk of misinterpretation is too great. Only data that are common to *all people* in the company ought to be stored in a common database. Another way of expressing this is saying that only intersubjective data are to be stored in a common database. Nurminen (1982) has pointed out the disadvantages of too great a degree of integration.

Since a support system mostly supports a single user in his or her work and since every person has a working style and an apprehension of reality of his own, the system must be, at least to a large extent, individual. Obviously we can not use a traditional systems development approach in developing such systems. User development appears to be an absolute prerequisite for a good support system.

Functions of such a support system might be:

1. Database operations for storing and retrieving data.
2. Calculation operations in order to perform mathematical manipulations of numbers representing some sort of model of the company.
3. Graphics operations in order to represent a certain set of numbers as curves or bars or pie charts.
4. Word processing to support writing of memos, reports, letters, etc.

Two other operations might also be considered:

5. Mail systems in order to exchange written messages with other users maybe far away or at times that suit the involved persons.
6. Information retrieval in external data banks.

Today there exists a variety of tools more or less suitable to perform these operations. Since these tools have been in use for a number of years already it would be an interesting research task to investigate the effects when using such a tool.

Scientific Approach

A traditional approach to such an investigation should be the verification of a set of hypothesis. This set must however be based on an understanding of the factors that are important. You do not get answers to questions that are not asked! The factors are parts of a theory, so we thus need a theory of user developed support systems in order to undertake a traditional scientific approach.

This theory could deal with the tools that are used and how they should appear and behave. We then have a technical knowledge interest, according to Habermas (1978). The theory could deal with effects of support systems and try to explain why these effects have arisen. Thus we have an explanatory knowledge interest (Habermas 1978). The theory could also deal with possibilities for people to improve their work using support systems. We then apply an emancipatory knowledge interest (Habermas 1978). The choice of knowledge interest is not a scientific question, it is a political or ethical choice. I

personally do not think it is morally right to prescribe for other people how they should work in detail and I also think work should be designed in order to achieve job satisfaction for people doing the work and not simply to achieve high performance. Thus I basically apply an emancipatory knowledge interest.

However, I cannot avoid using the other two also, since I know too little about the empirical experiences of this type of systems use and development. In order to gather these experiences, I must apply an explanatory knowledge interest. I also think the tools used for developing and using support systems are important to study, since every tool has a basic idea behind it. A query language, for instance, might require a common database. This, as already shown, partly violates the idea of personal support systems.

Thus I must use two different approaches in the study of user developed support systems. The first one is based on collecting experiences from such systems and documenting them in case studies, the other one is based on taking an active part in such developments. I will now consider certain methodological issues in these two approaches.

Case Studies

Collecting experiences means in practice asking a lot of people about their experiences. I must have some ideas about what might be interesting but it is also important to allow people to talk about other things. Interviews with a few rather broad questions and possibilities of asking more detailed questions seems to be an acceptable approach. Questionnaires are probably not a valid technique here, since the purpose is to let the people concerned talk about their specific experiences. I think case studies are a better way of reporting this type of research. Perhaps some common pattern might emerge and then, but not until then, the specific pattern could be investigated using harder data gathering techniques.

The problem is how to find the broad questions. I could, of course, ask only one thing: Please tell me about your work with the computer. In fact I have started my interviews with exactly that question. Often people then talk about how it all began. However, I think my interviews, should reflect my understanding, so far, about the area of investigation. Therefore I put forward some areas I thought ought to be of interest, such as:

a. The tools that are used
b. The organization that uses the tool
c. The historical background
d. A description of present application
e. The effects on job satisfaction
f. The organizational effects

These are the main areas I try to cover in semi-structured interviews with individual people. In order to make the so far very vague theory explicit, to both myself and other researchers helping me in the investigation, I developed a set of questions covering all six areas above. I have in total 84 different questions that could be asked. However, and this is very important: *It is by no means my intention all these questions should be asked as a mechanical questionnaire!* When I do the interviews myself, I have found that it is usually sufficient to ask the main question in each area, and the details concerning the

other questions are answered without my having to ask them directly. I tried to maintain a conversation, rather than making an interview. That is another reason for why I called the technique "semistructured interviews."

The data processing department is supposed to answer questions concerning the tool, at least the more technical ones. I think, however, it is important to ask the users too, because in using the tool they might have gained this kind of knowledge, perhaps at a higher degree than people from the data processing department. It is also possible to compare the description of the use of a certain tool, as it is described by the vendors, and the description of its actual use.

Concerning the organization that uses the tool, I have so far only considered fairly large organizations, which already have a working data processing department. The idea of user development has often come via the concept of the "information center" as it is suggested by IBM. Mostly only a few people in the data processing department are working with user development. I also intend, however, to study small organizations and their way of working with this type of computer usage.

The historical background is very important. It helps me better understand the phenomena I come across in the investigation. However, different people give different interpretations of what happens and why it happens. I also might come across integrity problems, which of course must be respected. Sometimes people do not know much about the story prior to their appointment, since they were employed to work with the computer system and thus they had not joined the organization until the system was in use. In this case, I am forced to use second hand information.

I have used the framework of Mumford (1983) concerning job satisfaction and asked people about effects on their job skill, their social relations, their psychological needs, their ethical dilemmas and their total job load.

An important organizational issue is the possible support for decentralization in the organization. This can be one organizational effect. Another might be conflicts between data department or user groups or information center people.

The set of questions have been worked out in two steps. First two of my students did investigations in two companies, without any question set. Their conversations with the people in the companies were written down almost verbatim. I thus obtained a good impression of what took place. A certain pattern emerged: historical description, actual application, properties of the tool and organizational effects were the main points. However, people did not say anything about the effects on job satisfaction. Perhaps they were unaware of them, so I supplied a set of questions relating to this area. Then I made a test interview myself and found the total set of questions useful. I also prepared a guide with a more detailed discussion of every question and explained my intention with the questions and the type of answers I expected. I also tried to foresee the difficulties and identify possible ways of overcoming them. The purpose of this guide was twofold:

1) To act as a guide for other researchers or students that should help me in the investigation.
2) To document my own thoughts before the real data collection started.

A comparison of these thoughts with subsequent ideas might perhaps be useful in order to detect how I was influenced.

The interviews were conducted using a tape recorder to avoid having to take notes which might have inhibited the interview. Thus only reference to the prepared set of

questions was necessary. In conclusion, the interviewee was provided with the set of questions to check if any important or interesting questions remained unanswered.

Currently I have two levels of documentation, firstly the tape recorded conversations and secondly the conversations as subsequently written up in a coherent form. This requires some interpretation of the collected material. A typical conversation took about 1 to 1.5 hour and occupies about 8 to 9 full pages of text when written up. When I had conducted the conversation, I tried to write them up immediately. Some conversations have been done by students or other researchers and they were unable to do it this way; for example, some conversations have been done at companies far away, so people had to stay at an hotel over night. The written up versions are sent back to the interviewed people for comments and corrections. Then I called back after about a week to check if my documentation was correct and to see if any of the set of questions were unanswered. When they have agreed, the written conversation becomes an official version and could be distributed. This follow up has proved difficult as people are always very busy.

The material has to be confidential as some interviews reveal information of sensitive nature that cannot be published or alluded to. I thus have to write in such a way that the actual applications are only very briefly described and the company and the products are not mentioned by name. In doing this, I hope to guarantee the integrity of the organizations involved.

I also try to have two different researchers write up conversations from the same tape and compare them. Thus I can, to some extent, determine the influence of the researcher. The primary material is the written conversations, not the tapes, and thus I must assure a high quality and a close match with "reality."

Some Findings

At present I have nine conversations from four different companies and some patterns are emerging.

The first finding is the existence of *the local experts*, that is, people who know how to use the tools and help other people in *their* work. These local experts work with the computer tools as a part of their job. They frequently have a relatively high position in the organization, working close to the head of the department or maybe the company. In one case, the local expert was the head of the purchase department. They were all enthusiastic about the possibilities this new way of working provided. They often become more influential and felt more valued by the head of the department or the company.

Another finding is that they use more than one tool. Typically there were tools for databases (query languages), calculations, graphics and word processing. The tools are supplied by an information center at the data processing department. At this center, two or three persons works as advisers and educationists concerning the use of these tools. APL is often used as a host or base language. This reflects the fact that IBM uses this in their information center idea.

Irrespective of the tools they are using, the people were satisfied with the choice and the way of working with them. On the other hand, they do not seem to perform any deep evaluation either of tools or of suitable applications. The choice seems to be more or less random. No company had any well planned strategy for introducing this new way of

working. But the people using the support systems were all convinced about the profits, both in money terms and particularly in terms of quality of work performance. They all thought they could do a much better job than before and also do things they had not previously conceived of doing.

I have found it relatively easy to detect effects on job skills since the people without exception so far think the use of a user developed support system have greatly improved their ability to do a good job. Concerning the other job satisfaction factors, no effects have been detected. Some people admit they have more influence due to their improved job performance, but that is all.

I also distinguish two or maybe three levels of using the tools. The first level is the occasional user who uses a prewritten procedure in order to get a report from the database or some computations in the calculator system or a picture plotted with the graphical system. (S)he also might use the command mode of the query language or the calculator in order to answer some ad hoc questions. The second level is the local expert, who can write procedures, create new files and help other people with more complex questions. The third level is perhaps people from the information center who help the local expert with the more complex jobs or when the system has broken down.

People use the support system to produce reports, which they are working with at their desks. The terminals are not necessarily located on users' desks, but can be found in another room, enabling more people to make use of it. However, people working with large calculations seems to require the terminal at the desk.

Response times fluctuate widely. I have seen applications with surprisingly quick response times but also applications where response time could be up to 15 minutes. As a rule, people think they are rather slow, but it is not considered to be a serious problem.

Some people use microcomputers and some use terminals connected to a central mainframe. In one company, they first used micros but then switched to terminals and the terminals were considered to be much better. This was due to the mainframe in addition providing a set of tools very much like those of an ordinary micro.

Knowledge of the data model (the system) was more important than knowledge of the tools. The databases used by the companies were usually complex and required great skill from the users of the support systems to make effective use of it. Thus the local experts were experts both on the tool and on the data model. Perhaps they could be considered as local database administrators.

Critique of the Methodology

The purpose of the research is to get a brief overview of empirical experiences and to describe what people think about their job with user developed systems. Some common patterns emerge, but can by no means be considered as traditional, positivistic hypotheses that have been verified. Perhaps such hypotheses can be formulated and experiments or investigations designed in order to verify or falsify them. If this is the case, then I shall be very satisfied with the investigation.

Since the interviews are quite free (rather conversations), I hope the most important effects and experiences from the users' point of view will emerge. However, people see me as a computer expert and they often think I require certain answers. In such cases they

often unconsciously give me just these answers. This "counter-interview" is hard to avoid. One way might be to remain as silent as possible. It is however impossible to make any comparative experiments in this area since every interview situation is a unique one.

Another important question is how do I know that my six main areas are the important ones. They are only a structure, I try to collect and report the data in such a way that I am not restricted to these areas. But of course you cannot get answers to questions not asked. I intend to make an in-depth evaluation before I continue the investigation. Perhaps I shall discover some shortcomings in my structure. Presentation to other researchers is a part of this process.

I have considered individuals all the time, not organizations. This orientation is perhaps disadvantageous but my reason is I think not many investigations from the individual perspective have been undertaken. It is, however, possible to compare people working in the same organization and perhaps find some common patterns due to organizational factors.

Action Oriented Approach

The case studies describe the phenomena of user development from the observer's point of view. It does not provide any inside or in depth knowledge of effects that might occur during the implementation of user development in an organization. I think that process, too, must be documented, in order to avoid pitfalls and gain knowledge about the user development process. The question of for whom I gather this knowledge, and which Nissen (1984) discusses in his paper, is, in this case, a very important one. In the case study part of the project, I produce rather generalized knowledge for "society" or something that perhaps resembles the community that Apel identifies (Nissen 1984). In the second part of the project, which I call the action oriented approach, my role as researcher is not so clear cut.

I could choose the role of observer only and document what happens without any intervention. This is suitable when I know nothing about the problem. But in this case, I do. I have gained some knowledge from the case study part of the project and I have a strong belief that the users should have great influence. I believe it is wrong and ethically unacceptable to act only as an observer and not try to help people involved in the user development. This is however not a scientific approach from the traditional scientific point of view. It is ordinary consulting work. My main problem of scientific legitimation is thus to draw a boundary between consulting and science. I think that the knowledge interests and the community of Apel (1980) could help me in doing this.

A consultant has a clear commercial interest. A researcher ought not to have that. This can be expressed in the Chinese saying: "Give a man a fish and he has enough to eat for that day, teach him to fish and he has enough to eat for life." I should thus act as a teacher and adviser and teach people how to develop systems rather than do the job for them. However, some consultants or system analysts act in this way too, an example being those working in Information Centers. The difference, I believe, is that scientific knowledge should be common and distributed to everybody. This is, as far as I can understand without having read Apel in detail, the main idea of the scientific community. This makes a heavy demand on the researcher. One implication is the need to report and

even research in such a way that every member of this community can understand it. As far as I can see, the implication is that I must be able to explain my research at any time in the research process to anybody that is interested!

The possibilities for achieving generalized results are almost zero, but the possibilities of achieving valuable knowledge for the people involved in the project are enormous! I have tried to act as a change agent and obtain acceptance for the idea of user developed support systems in the organization. Often I felt as like a salesman of ideas. I believe my ideas to be good for the organization, but I cannot prove it.

Another problem is identifying what the result of the research actually is. Usually research reports are considered to be the main result. In action research projects, however, actions taken by participants in the project might be considered as results of the research. Nygaard (1974) has described this dilemma in the NJMF project. However these actions and interventions must be described and reported somehow. I think it is a fair demand from my research colleagues. My research cannot be reproduced but it can be described although the description must not reveal any trade secrets or other sensitive information. It is a problem of integrity, both for the company as such and for the people involved. Trade secrets are not to be gathered or documented in any way. If I discover such information, I rewrite it in very general terms or disguise it. I sometimes have to document personal and sensitive information in order to explain a certain intervention or a specific result. In my own, internal documentation, this information can appear, but in the official documentation it must be transcribed or generalized.

Some Results

My work in the company as an active participant in user development can thus be reported as a plain description of what happened and perhaps of my thoughts at the time. Some background facts are however necessary. The company is a very old company. It was founded in 1797 but its roots are further back in the 16th century. The company has had problems in the later years. Recently almost half of the white collar staff have had to leave the company. A whole division was sold. However, things have begun to change and the result of 1983 was good enough and for 1984 it seems to be even better.

The problem concerned a financial system from which it was rather difficult to obtain desired information. The information existed in the system, but it was impossible to get at it. The system is run as a batch system and producing lots of lists. It was developed as an in-house system by the data processing department. It is a centralized system supplied with information from a couple of presystems. These presystems are run on different computers, the communication with the central IBM 4331 being via diskette and tape exchanges.

In the spring of 1983, a project group was formed consisting of the present head of the data processing department, two people from the central finance staff and two researchers from the University of Lund. One of the latter was myself and I was considered as scientific project leader. The project leader for the company was the head of the data processing department. This changed about Christmas, when the head of the data processing department left the company and another person from the central finance department replaced him. Also the other researcher was unable to continue to participate.

The project started as an ordinary systems development project. In the autumn, we undertook a problem investigation and a brief description of the company. I tried in vain to understand their finance system. The researchers and the data processing head tried to obtain a requirements specification from the other two finance people, but this was also in vain. At Christmas I suddenly realized I was acting as a traditional data processing expert and not as a user oriented data researcher! I made a description of my own point of departure from this role in a planning report. In the company's view, this was an unusual and embarassing report. I realized this and tried to adapt as much as possible to the behavior of the company. Thus I hoped to gain their confidence and perhaps acquire a more favorable response for my own ideas.

After Christmas things began to work much better and it became clear that it was a support system we were going to develop, not a control system. I also realized it was impossible to produce a traditional requirements specification of the desired system. In fact this directly led me to develop the concepts of support and control systems and the realization of the impossibility of specifying office work.

I found four different ways of developing requirements specifications. The first and the easiest was to identify the inadequacies of the present system. We used this method at the beginning but not systematically. The second was the traditional one of asking what demands and information were required of a future system. This was, as already said, impossible. The third was to describe the work, then analyze which activities could be performed by computers and then find what information was needed for these activities. I have in previous research projects (Flensburg 1981) developed a method for doing such an analysis. This was also impossible since the work of the people was not describable. The fourth was to find the data structure of the existing presystem and then determine the data needed. This method was suggested by one of the users in the central finance department.

1 tried making the future system visible for the users (and myself) by constructing some simple prototypes with DBase II on a microcomputer. As a result of this, we obtained an apprehension of a possible requirements specification. I also ran a short course on different data models for people in the data processing department. Finally we managed to develop a requirements specification by looking at the data descriptions for the existing presystems and chose the data terms we wanted. I translated these to a data model in the third normal form. It was our intention to use this model as a test case when we visited vendors, selling tools for development of support systems. We visited some vendors and the tools were demonstrated. However, after a couple of demonstrations we were more confused than helped.

It also turned out that the investment needed to be rather larger than anticipated and thus a decision needed be taken at a higher level in the organization. The process of convincing management that this investment is worthwhile is still in progress and a decision is not expected for some time. The company has a policy of buying computers from only one vendor (not IBM!) and we need to investigate the tools this vendor offers.

Critique of the Methodology

The great problem with this type of research is to keep the primary material to a manageable size. It is not possible to know, in advance, what is important or not.

Everything can be important and thus perhaps I should have recorded everything that happened on video tapes. This would be extremely expensive and probably impossible to use. I tape recorded (on audio tape, not video tape) certain meetings and documented them afterwards in written form. Some meetings were only documented in my own notes and some were not documented at all, examples of the latter being some visits to vendors for demonstration of user development tools. The written documents are distributed to the project members as "meeting notes." In some cases I have produced planning reports or other more traditional reports intended for use by the management of the company. I also keep a private logbook, where I put personal and speculative thoughts. This is intended for my own private use and might serve as background for "meeting notes" or other reports.

The criteria for failure or success are also very important. Quite by chance a good success criteria evolved. One of my fellow researcher complained about people not giving her credit for her ideas. She said: "I suggested something two months ago that they are now using and they do not even mention my name now, but act instead as if it were an idea of their own!" In my opinion this is a very good success criteria. We are not working with people in order to get credit for our ideas, we are working in order to help those people. If they take one of my ideas as their own it is surely a very positive indication of our success.

Comparison of the Two Methodologies

When two methodologies are to be compared, you must use a method of comparison that is independent of the compared methodologies. In this case I have chosen a comparison based on the knowledge interests of Habermas (1978). The case studies are mainly based on an explanatory knowledge interest. I describe what happend and why the people involved thought it happend. When I have more case studies (I think about 20 or 30), perhaps a more significant pattern might appear. This could be used for traditional hypothesis formulation. There are basically no contradictions between positivistic methodology and my case studies. They can very well be considered as two branches of the same tree, the difference in my approach being perhaps the careful investigation before the formulation of the hypothesis.

The knowledge interest of the action oriented research is mainly emancipatory. I want the people in the company to realize the benefits of user development and use of tools for that purpose. The knowledge that might be of more general interest is the process of achieving user development, what obstacles are there and how might they be overcome. This knowledge can, for obvious reasons, not be used for generation of hypotheses, but it can be used for generation of methods and approaches. For instance, we have found a need for methods of choosing and evaluating different tools for developing support systems. Methods for this will be both developed and evaluated by the members of the project.

The action oriented method has however a severe shortcoming. I have only one case! Therefore I have found at least two further cases. One is the employment exchange for the region. They have a couple of micros and a lot of problems that could be solved with the help of them. They also have software for doing the job (DBase II) but no person to

help them. Some students have developed four different systems for these people during the spring of 1984 and I intend to follow it up during the autumn. Of particular interest is the users' opinion of the developed system and the effect on their job and job satisfaction. One such effect seems already obvious: The man who is supposed to take care of the systems shows serious symptoms of computer hacking!

In another area of local government, they intended to introduce a new salary system. Fortunately one of my fellow researchers of the LUIS group (Siv Friis) became involved in that project at a very early stage. We have collaborated and held a course entitled "How to put requirements on data systems" for all the people in the salary department. It was very successful and the management now want us to continue as support during the development of the new system.

The whole research process can be described in the following figure, which is a modification of a similar figure in Checkland (1981, p. 8). I have not so far discussed the question of whether user development takes tacit knowledge more into account than traditional development. But the truth is, none of the applications considered here could have been developed in a traditional way. The data systems are used in quite a different way than is supposed in the traditional way. In the applications I have studied, the people use the computer as a real tool, comparable to an electronic calculator (in fact, one of the interviewed persons did exactly that). In traditional data systems, people are either humble servants of the systems (data feeder) or passive receivers of thick paper lists (data needer) (cf Sandström 1984).

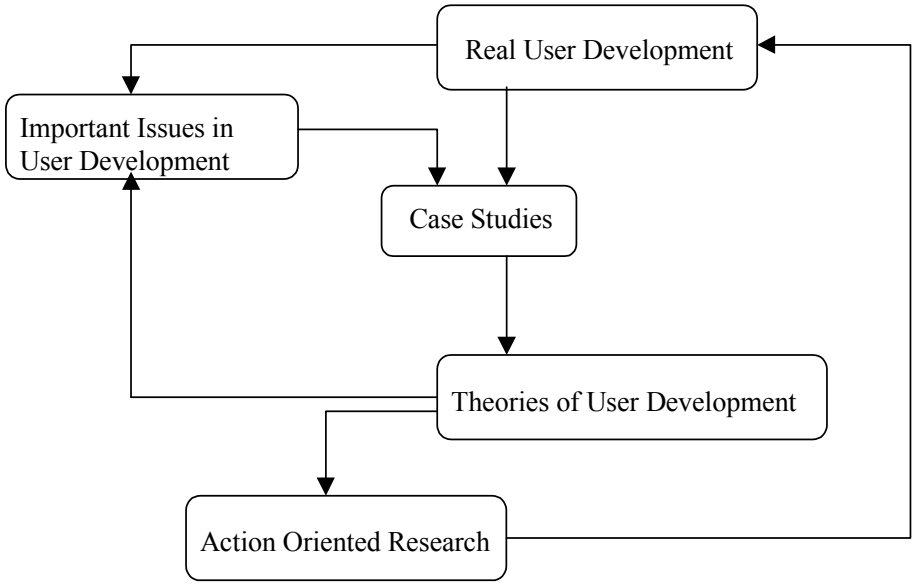

Figure 1. Relationships between the Different Parts of the Research Project

References

Apel, K. O. *Towards a Transformation of Philosophy*, London: Routledge & Keagan, 1980.

Checkland, P. *Systems Thinking, Systems Practice*, London: Wiley & Sons, 1981

Cooley, M. *Architect or Bee?* Slough, England: Langley Technical Services, 1980

Flensburg, P. "Towards User Development: A Dialogue," in *Report from the Fifth Scandinavian Research Seminaar on Systemeering*, Kerola and Koskela (eds.), Oulo, Finland, 1981

Goldkuhl, G., and Lyytinen, K. "Information System Specification as Rule Reconstruction," *HUMOR*, 1983.

Gödel, K. "Uber formal unentscheidbare Siitze der Principia Mathematica und verwandter Systeme," *Monatshefte fur Mathematik und Physik* (XXXVIII), 1931, pp. 173-198.

Göranzon, B. *Datorn som verktyg*, Lund, Sweden: Studentlitteratur, 1983.

Habermas, J. *Knowledge and Human Interests*, London: Heineman, 1978 (German original: Suhrkamp, 1968).

Langefors, B. *Theoretical Analysis of Information Systems*, Lund, Sweden: Studentlitteratur Auerbach, 1966.

Lindholm, S. *Vetenskap, verklighet och paradigm*, Stockholm: AWE/Gebers, 1979.

Martin , J. *Programming Without Programmers,* Englewood Cliffs, NJ: Prentice-Hall, 1982.

Mumford, E. *Designing Human Systems*, Manchester, England: Manchester Business School, 1983.

Nissen, H-E. *On Interpreting Services Rendered by Computerized Information Systems*, Dissertation, University of Stockholm, 1976.

Nissen, H-E. "Acquiring Knowledge of Information Systems: Research in a Methodological Quagmire," Department of Information and Computer Sciences, University of Lund, 1984.

Nissen, H-E.; Carlsson, S.; Flensburg, P.; Holmberg, K. A.; Sandström, G.; and Wormell, I. "User Oriented Information Systems: A Research Program," Department of Information and Computer Sciences, University of Lund, 1982.

Nurminen, M. "Against Systems," in *Report from the Fifth Scandinavian Research Seminaar on Systemeering*, Kerola and Koskela (eds.), Oulo, Finland, 1981.

Nurminen, M. "Human-Scale Information Systems," Institutt for Inform asj onsvitenskap, University of Bergen, 1982.

Nygaard, K. "Fagbevegelsen - en ny opdragsgiver," in ... *over til ED*B, R. Høyer (ed.), Oslo, 1974.

Polanyi, M. *The Tacit Dimension*, New York: Anchor Books, 1968.

Sandström, G. "Improved Retrieval from Information Systems," Department of Information and Computer Sciences, University of Lund, 1984.

Tarski, A. "Der Wahrheitsbegriff in den formalisierten Sprachen," *Studia Philosophica* (1), 1935, pp 261-405 (Polish original, 1933).