PROGRAM DEVELOPMENT AS A SOCIAL ACTIVITY

Kristen Nygaard
Institute of Informatics, University of Oslo,

This paper was given as an invited lecture at the Dublin 1986 World Congress of IFIP (The International Federation of Information Processing Societies). It has served as a summing up of the work I had done in system development research together with colleagues and team members up to the mid 1980s. Since then, I have professionally mainly been working in object-oriented language design. Politically, the years 1988-1994 were consumed by my leadership of the "No to EU"-organisation till our victory on 28 November 1994 when Norway said No to membership in the EU.

I am now trying to integrate my work in different areas of informatics into a unified approach to the science. (See the Activity "Informatics and Research") This has resulted in a number of modifications and more precise versions of many concepts. Some will be given here, as annotations to this paper.

A more complete revision will be a part of a larger paper that I am now working on.

Reference to this paper: Nygaard, Kristen, 1986: "Program Development as a Social Activity", INFORMATION PROCESSING 86, H.-J. Kugler (ed.), Elsevier Science Publishers B.V. (North Holland), IFIP, 1986 (Proceedings from the IFIP 10th World Computer Congress, Dublin, Ireland, September 1-5, 1986), pp. 189-198.

Preface

This paper was given as an invited lecture at the Dublin 1986 World Congress of IFIP (The International Federation of Information Processing Societies). It has served as a summing up of the work I had done in system development research together with colleagues and team members up to the mid 1980s. Since then, I have professionally mainly been working in object-oriented language design. Politically, the years 1988-1994 were consumed by my leadership of the "No to EU"-organisation up to our victory on 28 November 1994 when Norway said No to membership in the EU.

I am now trying to integrate my work in different areas of informatics into a unified approach to the science. (See the Activity "Informatics and Research".)

As a part of that work, I am reexamining my earlier work, revising sections where I either have changed opinions or feel that my insight has improved, or where I feel that I have more precise and consistent definitions and concepts to offer.

This paper is of course given in its original version. My revisions vil be made as annotations to this Acrobat- and WWW-adapted version of the paper for the Internet.

The work on the annotations has not yet begun. Hopefully it will start soon, and it will go on throughout my work on a "Unified Approach to Informatics".

Oslo, 99-08-16

Kristen Nygaard

1. INTRODUCTION

This paper will argue on two fronts: The first front will be against those who want to make informatics exclusively a formal science, as mathematics - and who maintain that the study of the social implications of information technology is outside informatics, belonging to the social sciences. The second front will be against those social scientists who believe that they are capable of understanding the impacts of information technology "from outside", without bothering to learn informatics or working closely with researchers in informatics

The paper will not be laid out as a series of attacks, but rather as a presentation of points of view, arguments, and lines of reasoning that are in in contrast with the two attitudes described above.

In recent years a comprehensive litterature has addressed the technical and organizational problems and tasks in large scale program and system development. *[The list of relevant litterature is far too long to be included in the Reference List, see e.g. [Boehm, B.W., 1981], and Olle, T.W., Sol, H.G., and Verrijn-Stuart, A.A., 1982].]* The paper will use that litterature as a background, but will focus upon the interaction between technical and social issues in these activities.The paper will, however, to a large extent draw upon points of view presented in papers by researchers (including the author) belonging to the Scandinavian trade union and user participation oriented school in System Development.

It should also be pointed out that the term "social" in the title of the paper will mainly be used in its meaning of "societal", that is, relating to the society. This implies that the social setting of the programmers' work wil be discussed, but not the internal relationship and cooperation in project groups of specialists in informatics.

2. THE DEFINITION OF THE TERM "INFORMATICS"

The term "computer science" should be replaced by "informatics". Some years ago the choice between the two terms seemed perhaps to be of little consequence. Discussions of terminology often are idle, but they may some times reflect basic differences of opinion or at least emphasis. Today it is unfortunate that the term "computer science" is used. The term tends to implant a too narrow way of thinking about information systems, that now are networks of people, information processing equipment and other machinery, interacting through direct inter-human and (an increasing proportion of) electronically supported communication links.

The definition of the term "informatics" adopted in this paper is *[rephrased from [Nygaard, K.and Håndlykken 1981]]*:

Definition: Informatics is the science that has as its domain information processes and related phenomena in artifacts, society and nature.

In this definition, the term "information processes" is used. "Information processes" are discussed later in this paper.The definition includes the term "information", and that term raises another debate. Many attempts have been made at giving a definition of "information". An approach using the perspective and information process concepts is presented in *[Nygaard, K., 1986]*, another is given in *[Kaasbøll, J., 1985]*.

A phenomenon is

" any fact, circumstance, or experience that is apparent to the senses and that can be scientifically described or appraised." *[Webster 1960]*

Important examples of phenomena are: living organisms, inanimate objects (including artifacts, like e.g. machines), events, and processes (e.g. computer program executions). We may also speak of cognitive phenomena, occuring within the minds of people, as opposed to manifest phenomena, occuring outside minds.

The above definition describes informatics as a science relating to a class of phenomena (and certain selected aspects of these phenomena - see the definition of information process below). It is incompatible with definitions that aim at defining informatics as a formal discipline, akin to mathematics. (Formal sub-disciplines may, however, be parts of informatics - as e.g. program proving.) The definition is similar to those we must give of sciences like botany, physics, sociology etc., which all relate to selected aspects of specified classes of phenomena.

The choice of definition of the term "informatics" is in no way trivial. When it is argued that informatics is a formal discipline only, then

"according to such a definition, the impact of an information system upon the social structure of which it is a part, is outside (the field of study of) informatics. Also case studies of how data processing actually is carried out in specific organizations fall outside informatics in this narrow sense." *[Nygaard, K., and Håndlykken, P., 1981]*

Terms used incorrectly may also cause confusion. It has been maintained *[A. Parkin 1980]* that "informatics" is a substitute term for "data processing". "Data processing" is, however, a term denoting a process or a function, whereas "informatics" is the name of a science that has data processing as a subject of study.

3. ASPECTS OF SCIENCES

Sciences defined the way informatics is defined above, have four aspects [The material in this section uses material from [Nygaard, K. and Sørgaard, P., 1985], developed from [Nygaard, K and Håndlykken, P., 1981]] :

1. *Phenomenology:* The empirical study of phenomena - their identification, observed behaviour, and properties. (Tycho Brahe in astronomy, Linné in botany.)
2. *Analysis:* Comprehension and explanation of phenomena in terms of an underlying theory. Identification of important properties and concepts, relations between properties and concepts, description and anticipation of behaviour. (Newton in astronomy, Darwin in biology.)
3. *Synthesis, construction, technology:* Knowledge organized for the purpose of interfering with, constructing, or generating phenomena.(Teller in nuclear physics.)
4. *Multiperspective reflection:* The consideration and examination of concepts and phenomena at the same time - or alternatingly - from the perspectives of more than one science, or from more than one perspective within the same science. The study of how changes introduced according to one viewpoint affect properties of the phenomena when regarded from another viewpoint.

Most natural sciences now have long phenomenological traditions. Many, like physics and chemistry, have developed technological disciplines built upon solid platforms of analysis, which in turn are derived from very extensive platforms of empirical knowledge. (Some, like chemistry, started however with rather unsuccessful attempts at construction.)

Informatics developed as the result of the invention of an extremely powerful tool for construction: The programmable computer.The very mixed crowd of people gathering around the computers were regarded by scientists (and often by themselves alike) as "machinists" or "technicians". During the 50s the first generally valid insight related to construction was established. Analysis was firmly established in the 60s, whereas more explicit multiperspective reflection started to appear in the 70s, in particular through the Scandinavian trade union projects. As for phenomenology, valuable efforts have been made, but too few, and also many made in order to prove that some proposed method works successfully. Informatics need in a number of important areas a proper and wider empirical platform.

It seems obvious that construction always will be a central, or the central aspect of informatics. Without construction the purpose and content of the other aspects become empty. Information systems are, however, typically introduced and operated for economic reasons, designed and implemented using insights from informatics, and result in a changed environment for people. Modifications of the informatical properties of the organization may have desirable economic, but undesirable social consequences, or vice versa.

4. PERSPECTIVES

The "perspective" concept is used in the previous section. It is a key concept in the discussions presented in this paper, and it will now be examined more closely, following *[Nygaard, K., and Sørgaard,P., 1985]*. The perspective concept is also discussed in *[Bratteteig, T., 1983]*.

As our first illustration we will refer to a very typical situation:A system analyst is working in team with Ms. Brown and Mr. Smith (representing "the users"). Their task is to develop some kind of new "information system" in the organization where Ms. Brown and Mr. Smith are employed.

To most system analysts it is a matter of course that the organization should be interpreted as consisting of "data flows", "transactions", "records", "relations", "objects", etc. This is their perspective upon the organization. Rather few are capable of realizing that to Ms. Brown, Mr. Smith is not a system component. Ms Brown has a very different perspective: To Ms. Brown, Mr. Smith may be a colleague who should be treated rather carefully before 10 am., who is very knowledgeable and helpful with some of the tasks that Ms. Brown find difficult, who needs Ms. Brown's help with some other important tasks, and is an altogether pleasant chap, making up a good problem-solving team with Ms. Brown. The system analyst should realize that this way of looking at the organization may be just as relevant as a "system view", and should make an attempt to adopt both views.

To Ms. Brown, her workplace is not "a system", whereas the system analyst regards the purpose of the work she/he is engaged in exactly as: to develop "a system". The implications of adopting a system perspective are far reaching if alternative perspectives are neglected or outright suppressed. The Norwegian sociologist Stein Bråten has introduced the notion of "model power" *[[Bråten, S., 1973] and [Bråten, S., 1983]]* which in our terminology should be described as "perspective power". The system analyst exercises perspective power and establishes a perspective monopoly by insisting upon, and succeeding in the exclusive use (in the development process) of facts, experience, concepts, techniques, and tools that are meaningful within the framework of a system perspective. According to Bråten the acceptance of another actor's model (perspective) implies that one is completely dominated.

In philosophy concepts similar to "perspective" have been discussed by many researchers and are in focus in the theory of knowledge. The discussion here is oriented towards issues within the science of informatics.

Using the concepts in the previous section, a definition of the term "perspective" will be given. It is necessary to make some assumptions relating to cognitive processes (human thought processes). It has been attempted to make them as few and uncontroversial as possible.

Definition: A cognitive universe is the collection of all cognitions of a person.

The term cognition may denote either the process of cognition or any product of that process, in some way stored (or "remembered") in a person's cognitive universe. We do not make detailed assumptions about these products, only that we will regard both values, beliefs, experiences etc., and rules relating to these categories as being cognitions. We particularly do not assume that cognitions may be regarded as "disjoint elements", or that any part of a cognitive universe has a sharply defined boundary to other parts of a cognitive universe. Whether this is true or not is outside the scope of our discussion.

Definition: The active cognitions of a person in a given situation are those parts of the cognitive universe that are participating in the person's cognitive process.

We may now state :

Definition: A person's perspective is a part of the cognitive universe that may structure her or his cognitive process when relating to situations within some domain.

By the verb "structure" we in this context imply "contribute to the structuring of", since the complete structure in operation in a given situation is impossible to know.

Definition: A perspective structures a person's cognitive process by:
- the selection of those properties of the situation that are being considered (and, by implication, of those that are ignored). In this way the perspective influences the active cognitions.

- concepts and other cognitions that are being used in the interpretation of the selected properties.

If the use of the perspective is regarded by the person as "successful", cognitions reinforcing the belief of its usefulness is produced. If the perspective leads to a "poor" interpretation, new cognitive processes are initiated in order to modify or even substitute the perspective.

The capability of multiperspective reflection is essential for every computer professional, and the perspective concept will be used extensively in this lecture.

5. INDIVIDUAL AND COLLECTIVE PERSPECTIVES

The notion of perspective has been defined by characterizing a person's perspective, and this may be the best way to start a discussion of the perspective concept. This approach does not make it impossible to use "perspective" to denote attitudes shared by many *[Nygaard, K. and Sørgaard, P., 1985]*.

Perspectives may be communicated by perspective description or by people being exposed to specific situations from which they learn. When it is said that a system development method "has a harmony perspective" (see next section), it implies that a person A making this statement is of the opinion that any one using this method will structure her or his cognitive processes during the system development process by a perspective that person A regards a being a "harmony perspective". Many of the concepts we find indispensable in human communication, also in the social sciences, are of this nature. And informatics has aspects belonging both to the natural sciences and the social sciences. This does not preclude precise definitions of concepts like perspective and their use in describing e.g. group attitudes.

Some perspectives are highly individual: attitudes towards family and friends, towards various forms of art etc.

Other perspectives are shared by many: All over the Western World managers hold similar values relating to the way business is conducted, and these values are usually regarded as obvious to the extent that they are seldom discussed. Alternative values are often considered as "ideological", "political" or "extreme", as opposed to one's own values which are considered "objective" or "mainstream".

Similarly, workers tend to have the same way of looking at events within their factory. The common parts of their perspectives are probably only to a limited extent communicated, and to a larger extent determined by common experience.

6. PROCESS AND STRUCTURE

This and the two subsequent section introduces concepts relating to programming and system description. [The material in this section is building upon work related to the DELTA and BETA languages, see.the References]

Definition: A process is a development of a part of the world through transformations during a time interval. Structure of a process is limitations of its set of possible states (and thus of its sequences of possible states).

"The economic development of Norway in the 16th century" is an example of a process. It should be noted that "Norway", the substance developing, is not a process. The term "process" comprises the sequence of changing states. (Consequently, the use in e.g. the SIMULA language of the term "process" as a subcategory of "object" is unfortunate.) The most important information processes are, in our present context, program executions and data processing performed jointly by people and computers.

In most programming languages the transformations are prescribed by imperatives (and thought of as actions) in structure descriptions called programs. The computer imposes structure upon the program execution using the structure description (program) as structure prescription.

Definition: A process is regarded as an information process when the qualities considered are:
- its substance, the physical matter that it transforms,
- measurable properties of its substance, represented by values,
- transformations of its substance and thus its measurable properties.

7. SYSTEM DEFINITION

Another term that needs consideration is "system". Some definitions are too general and imply that everything is a system. The definition given here introduces explicitly the perspective implied and makes an immediate link possible to the concept "information process" and thus to concepts in programming languages [*Kristensen, B. B., Madsen, O. L., Møller-Pedersen, B. and Nygaard,K. 1983a, 1983b, 1985*].

Definition :A system is a part of the world that a person

(or group of persons, during some time interval and for some reason) chooses to regard as a whole consisting of components,

each component characterized by properties that are selected as being relevant and

by actions relating to these properties and those of other components.

[Holbæk-Hanssen,E., Håndlykken, P. and Nygaard, K., 1975]

According to this definition, no part of the world is a system as an inherent property. It is a system if we choose a system perspective. It may be regarded as being a system by us, if we at the time choose to do so. It may, however be useful not to regard that part of the world as a system. It may also be regarded as a system in many different ways, as we specialize our system perspective by the choice we make in the selection of properties to be considered.

8. THREE BASIC PERSPECTIVES ON PROGRAMMING

Within informatics there are many important but different perspectives, which may or may not be mutually conflicting. In programming, three major and different perspectives are :

- Function-oriented programming: The computing process is viewed as a sequence of applications of functions to an input, producing an output, which in its turn may be the input to another function etc. The programming language Lisp reflects this perspective *[papers by McCarthy, definition in [Marty, J.B., et al. 1978]]*. Other languages usually also contain constructs for expressing the applications of functions.

- Object-oriented programming: The computing process is viewed as the development of a system, consisting of objects (components), through sequences of changing states. The Simula languages were the first to present this perspective

  [Simula I *[Dahl, O.-J., and Nygaard, K., 1965], Simula 67 [Dahl, O.-J., Myhrhaug, B., and Nygaard, K., 1968]]*. Smalltalk is an important later example *[Goldberg, A., and Robson, D., 1983]*. Facilities for object-oriented programming have been introduced in Lisp (Flavors in Zetalisp, Loops in Interlisp), in Pascal, C and other languages.

- Constraint-oriented programming: The computing process is viewed as a deduction process, developing from an initial state, the process being restricted by sets of constraints and inputs from the environment, the information about the states being deduced by an inferencing algorithm. Prolog *[see e.g. [Clocksin, W.F., and Mellish, C.S., 1981]]* is an example of a constraint-oriented language : constraints may be formulated in first order predicate logic (as Horn-clauses), and inferencing is made by the use of Robinson's resolution mechanism. An example of alternative inferencing algorithms is found in *[Nossum, R., 1984]*. More examples will no doubt be worked out in the years to come.

All these three perspectives should be supported within any general system description and programming language in the future: no perspective will "win" as some people seem to believe. (For this reason the term "paradigm" should be avoided, since the meaning established by Thomas Kuhn for the term is that of a basic perspective within a science, irreconcilable with alternative paradigms.*[Kuhn, T., 1970]*)

Object-orientation should be available in a system description and programming language because of its capability of system modelling and linking with the systems' environment, and also for its relevance to knowledge representation. Powerful functional modes of expression should also be available, as well as Lisp's incremental execution mode and operations upon its own program. Finally, a language should allow the introduction of new kinds of constraints and easy collection - during the computing process - of information to be used in inferencing by a wide range of alternative algorithms.

These three perspectives are not only of interest to programmers. If programs and systems are developed in multidisciplinary teams, with participation of later users, then it is important to have available system description languages that allow for a wider range of description modes.

9. THE PROCESS/STRUCTURE HIERARCHY

In the informatics defin ition above it is referred to "information processes and related phenomena". Let us examine the "related phenomena" that span out a wider canvas of informatics.

Applying Lars Mathiassen's notions of process-structure relationships *[Mathiassen, L., 1981]* to the phenomena studied within informatics, we may identify the following process/structure levels:

(1) Process: The information process (e.g. program executions, data processing by people and machinery in offices).

(2) Structure (of process at level 1): The limitations imposed upon this process by computer programs, machine hardware properties, written and unwritten rules of human behaviour etc.

(3) Process (producing and modifying structure at level 2): The system development process, including programming as a partial process, that has the structure of the information process (or the modification of its structure) as its product.

(4) Structure (of the process at level 3): The limitations imposed upon system development by organization, existing knowledge, available resources etc.

(5) Process (producing and modifying structure at level 4): The process of learning within organizations, the research process, the adaptation of organizations to a changed environment.

Informatics should not be defined as being restricted to level 1 and 2 above. It is ample evidence proving that the properties of an information process (at level 1) is strongly dependent upon how its structure (level 2) was produced (at level 3) *[See e.g.Andersson, J.,1974]*. In order to be able to give information processes improved properties (at level 1) new development methods may be needed (level 4), requiring research efforts (level 5).

The above process-structure levels are another example of a set of perspectives. A specific action or other event in a system development process may be regarded as a part of the programming process or, alternatively, as a part of a learning process changing the capabilities of the organization. The process-structure levels define levels of perspectives and not disjoint levels of phenomena, and the science of informatics has to address itself to all these levels.

Since the processes of programming and system development also are social processes, these processes may not be properly understood unless social perspectives and knowledge from the social sciences are introduced. On the other hand, the processes are strongly structured by their informatical content. More than one perspective must be used in a discussion of their properties.

10. THE SYSTEM DEVELOPMENT PROCESS

We may approach system development in a descriptive or in a normative way.

We may develop concepts for describing what system development consists of and make it possible to describe system development independently of concepts related to particular methods being used. This is necessary in order to be able to evaluate specific development processes and compare methods.

Mathiassen proposes the use of five functions to characterize system development processes *[Mathiassen, L. 1981]* :

- inquiry,
- construction,
- change,
- decision, and
- communication.

The system development methods represent normative approaches. They propose structure that ought to be imposed upon specific system development processes. In *[Mathiassen, L., 1981]* the following framework is proposed for characterizing methods for system development :

1. An area of application : A type of system to be produced, a type of system development organization.
2. A perspective, consisting of assumptions about the nature of systems, organizations, the surrounding society and the purpose of the local organization.
3. Principles for organizing the development process, splitting it into partial tasks, assigning resources.
4. Techniques of work used in the partial tasks.
5. Tools used in the application of the techniques.

Very few, if any methods are described by their authors using this layout. The assumptions, attitudes and values constituting underlying perspective are in particular very seldom stated, and they have to be deduced.

Using Mathiassen's framework it is possible to characterize and compare particular proposed methods in terms of its approach to inquiry, construction, change, decision and communication, as well as particular program and system development processes that are observed. One result common in such analyses is that actual development processes do not confirm to the prescriptions given in the methods one attempts to apply. It is also common experience that those "phase-oriented" methods are unrealistic that prescribe e.g. an "Analysis phase" followed by a decision, then "Design" followed by a decision, and finally "Implementation". In practice inquiry, construction and change are parts of all phases in the development process.

Comparing methods in terms of their area of application, perspective, rules for organization, techniques and tools also is useful and necessary to understand properly the implications and assumptions of methods that are considered for practical use.

11. USERS: INTERESTS AND ROLES

In system development it is common to refer to those affected by information systems as "users". In most discussions about the system development process this term is far too vague and general. Statements concerning "user participation" and "user satisfaction" may have no precise meaning without a specification of which group within the organization these users belong to. *[The discussion of the user concept in this section follows [Nygaard, K.and Håndlykken, P. 1981]]*. The groups of users may be categorized according to many criteria:

- Kinds of work tasks carried out.
- Position within the decision system.
- Position within the power system.

Two useful perspectives are those of <u>functional roles</u> and of <u>interest groups</u> , respectively *[see points made by P. Naur in [Fjellheim, R., Håndlykken, P. and Nygaard, K., 1974], and also [Nygaard, K., 1975] and [Nygaard, K., and Fjalestad, J.; 1981]*.]

We get the first perspective by considering the <u>functions</u> to be carried out by people in a given system by describing their <u>functional roles</u>.

A <u>role</u> is then defined by a specified task or group of closely interrelated <u>tasks</u> (constituting a job), carried out by people in the development and/or operation of a system. <u>A person may act in more than one role at a time.</u>(This role concept differs from that commonly used in social psychology where a role is described by the expectations that a person will adapt to in a given position in an organization.)

This list contains some important role categories in relation to development and use of programs and systems:

- <u>Rulers</u>: control,completely or partially, the resources used in the system (and its development) and the selection of the basic objectives it is intended to fulfill.

- Managers: survey and direct the development and/or operation of the system. Local representatives of rulers.
- Operators: work in the system and are necessary for its proper operation. An operator may often serve as a manager for a subsystem of the total system.
- Customers: interact with the system from necessity or by choice. In direct contact with the system only in shorter periods of time.
- Bystanders: may be influenced indirectly by the system, by its consumption of scarce resources or by other social effects.
- Designers: design and describe the system's structure (usually) in cooperation with management.
- Programmers: work out in detail the prescriptions (programs and work procedures) used in the system. Participate in the implementation.
- Teachers: teach operators and others how to use the system.

It should be stressed that the roles specify functions and not persons. A person within the system will often enact more than one role, at the same or at different times. (A programmer may, and in the opinion of many also should act as designer.)

When the management of an enterprise decides it wants "user participation", it usually means that some of the functional roles become represented in some of the bodies within the development organization. This user perspective is important also to interest groups (see below), if and when a consensus or compromise about the system's objectives and development is arrived at.

The role concept is useful because it emphasizes the various kinds of experience needed and the demand for different kinds of system descriptions in the development process. An employee needs one system description when she or he as operator shall learn how to use a terminal, another when she or he as shop steward (interest group representative) wants to understand how the terminals influence the social contact network and the control of the production.

The second perspective is that of interest groups. People with the same relationship to the production processes will usually develop an understanding of common interests, group interests. They will organize themselves in "interest organizations" acting as "interested parties" to decision-making in the society. Associated with an "interest point of view" is often also a more comprehensive perspective, in the form of an "ideology" or "picture of the world" that influences the group members' attitudes to what are important properties of a system and guides their behaviour in a development process.

A group's interests in systems in an enterprise or a public institution may relate to a number of issues:

- What economic benefit will we (the group) get from the system? (As employees, middle or top management, as employers.)
- To what extent may we exercise control (power) over the system?
- How will the system influence the physical and psychological working environment?
- Will the social network in the enterprise be changed to our advantage or disadvantage?
- What is the relationship between the objectives of the system and the objectives we feel should direct our society?

Data shop stewards are elected to represent group interests, and all points listed above may be raised in negotiation within the framework of the data/technology agreements presented later in this paper.

12. HARMONY AND CONFLICT

System development in public and private enterprises are part of the development of the production forces in the society. Organizational changes are often associated with shifts in power between capital and labour, and are in fact often introduced with this as a main purpose.

Every method for large scale program and system development embeds a perspective on the nature of the relationship between capital and labour. Any person studying system development must be aware of these aspects of the situation in order to hold realistic opinions.

Different political theories present different perspectives. Up to 1967, when the Norwegian trade unions started taking an interest in the application of information technology, system development methods in Scandinavia were always presented without any reference to possible conflicts between capital and labour. They were regarded as "objective", "neutral", "professional", "apolitical".

In the text book developed in the first trade union project *[Nygaard, K., and Bergo, O. T., 1973]* it is stated, however, that: "..... in our opinion language, concepts, models and theories for organization, job content and society are reflecting the interests and ideologies of those who created these languages, concepts, models and theories" (p. 27). And: "We are instead building upon ways of thinking (models) in which one may include both common interests and unresolved conflicts of interests. In our opinion this is necessary, since the objective of actions from the trade unions is a change in the power relations in companies and in their organization." (p. 28).

A researcher who has treated this subject carefully is Åke Sandberg *[in own papers , e.g. [Sandberg, Å.,1975] , and in a joint paper with Pelle Ehn [Ehn, P., and Sandberg, Å.,1976]]* :

A main distinction is made between:

- the harmony perspective, and

- the conflict perspective.

The harmony perspective expresses the fundamental view that the relation between different groups and individuals is characterized by harmony and the absence of basic conflicts. Conflicts are interpreted as "disagreement" and "misunderstanding" and may be resolved by information and conversation.

In the conflict perspective the relation between the groups essentially is characterized as an "unresolvable conflict" of "basic interests". This conflict must be handled by confrontation, negotiation and compromise, resulting in the postponement of the next confrontation.

	Conflict perspective	Harmony perspective
Observed conflict	Manifest conflict	Apparent conflict
Observed harmony	Apparent harmony	Manifest harmony

Fig. 12.1

Adherents of the two perspectives will interpret the same observed reality in different ways, schematically illustrated in Fig. 12.1.

Usually the supporters of the conflict perspective are more explicit than the harmony supporters. Åke Sandberg has stated this as :

"Not to choose is to choose the harmony perspective."

Examples of other perspectives of interest are the "transaction cost" perspective (on organizations) *[Ciborra, C., 1981]*, the "application perspective" (on the use of computers in organizations) *[Bjerknes, G., and Bratteteig, T., 1984]*, and the "tool perspective" (on computers) *[Ehn, P., and Kyng, M., 1985]*. A collection of articles written from a non-managerial perspective is found in *[Fossum, E., 1983]*.

13. PARTICIPATION AND REGULATION

In some countries there are laws and negotiated agreements between the main organizations of Labor and Employers, regulating the program and system development process. In Norway, where this started, one has to take into account:

1. the legislation on co-determination in enterprises (1972).

2. the "General Agreement on Technological Development and Computer-Based Systems" between the Norwegian Employers' Federation (NAF) and the Norwegian Federation of Trade Unions (1975, with later revisions).

3. the "Act Relating to Worker Protection and Working Environment" (1977).

4. the legislation protecting the privacy of information about "legal persons", both individuals and enterprises Effective from 1980).

The first demand for a "Data Agreement" came from a local union (Askim Chemical Workers' Union) in 1973 as a result of the "Iron and Metal Workers Project", a research project on "Planning, Control and Data Processing, Evaluated from the Point of View of the Trade Unions" (1971-73) *[see [Nygaard, K., and Bergo, O.T., 1973 and 1975] and [Nygaard, K., 1977]]*. After long and difficult negotiations, an agreement was signed early 1974 between the union and its employer counterpart.

In 1975 a corresponding agreement was signed at the national level between The Norwegian Employers' Federation (NAF) and The Norwegian Federation of Trade Unions (LO), and later between the trade unions and the government as well as the municipalities in Norway. The agreements probably cover system development at more than 90 per cent of workplaces employing blue and white collar workers (and unionization is very high in Norway).

The agreement gives the unions the right to elect special "data shop stewards" that are trained to take care of their interests in relation to computer-based systems. According to the LO, there are (1986) ca. 1500 data shop stewards in Norway (population 4 millions).

Some excerpts will illustrate how these agreements strongly impact upon the social aspects of program and system development.

"The Norwegian Employers' Federation (NAF) and the Norwegian Federation of Trade Unions (LO) agree that this general agreement shall form the basis for the planning, introduction and use of technology and computer-based systems. ....."

The agreement comprises technology and systems that are used in planning and accomplishing the work, as well as systems for the storage and use of personal data. Personal data refer to all data that can be traced back to specific persons employed at the individual enterprises by means of a name or other identification code.

"The use of technological possibilities in the form of equipment and systems may be decisive for the development and existence of the enterprise. New solutions and systems may influence the employees' workplace and working conditions.

When this is the case, it is important that the new technology is not just evaluated on the basis of technological and financial factors, but also based on social considerations. This overall consideration forms the basis for the design, introduction and use of systems and new technology, e.g. through consequence analyses.

"The enterprise shall keep their employees informed through their shop stewards about conditions which are within the scope of the agreement so that the shop stewards may express their views as early as possible and before the decisions of the enterprise are implemented, ..."

"The information shall be provided in a clear form and in language that can be understood by people without special knowledge of the field."

"The main organizations recommend that, in addition to the shop stewards' representatives, the employees who will be directly influenced by the projects within the scope of the agreement should be involved wherever practically possible in the project work."

"If the employees at the individual enterprise wish, they may select a special representative, preferably from among the existing shop stewards, to look after their interests and to cooperate with the enterprise within the scope of this agreement."

"The representative(s) shall have access to all documents about equipment and programs within the scope of this agreement."

"Examples of such training (of data shop stewards) are courses in systems work and project administration, sufficient to develop the competence needed to participate actively in systems design."

"Collection, storage, processing and use of personal data shall not occur without due cause based on consideration for the operation of the enterprise. The individual enterprise shall clarify the type of personal data which the computer equipment shall collect, store, process and use."

The above (and corresponding) agreement(s) are the main instruments when the social aspects of program and system development are negotiated in Norway. The "Working Environment Act" backs these provisions by general rules of law, but the agreements are more flexible for everyday use. They also reflect the fact that the employees only get so much influence as their unions are strong and have built up the necessary competence.

From this Act it suffices to quote §12.3 :

"Act Relating to Worker Protection and Working Environment.

§ 12, Planning the Work

3. Control and Planning Systems.

The employees and their elected union representatives shall be kept informed about the systems used for planning and controlling their work, and about planned changes in such systems. They shall be given training necessary to enable them to learn these systems, and they shall take part in planning them."

The provisions of laws and agreement give employees in Norway more influence over large scale program and system development than in other countries. The main trends of development in business and industry in Scandinavia are, however, determined by other forces, working at the world scale, and not by local laws and agreement in countries with fewer inhabitants than the main cities in many larger countries.

14. PROGRAM DEVELOPMENT AND PROFESSIONS

In an increasing number of professions people are using their most advanced competence in interaction with computerized information systems. The attempts at creating useful knowledge-based systems will enhance this development.

This implies that the mastery of these systems becomes an integral part of the professionals' competence. If the development of professions shall be influenced by its members, they must be able to comprehend the systems, participate in their development and be able to modify them as a part of the day-to-day adaptation of their tools to the changing and developing nature of their work.

It will not be possible to keep local system development specialists' groups capable of handling such tasks. "Programming" in the sense of making system extensions and modifications should be made by the professionals themselves, whereas the basic design and the keeping of database security and communications standards must be the responsibility of specialists in informatics or of multi-disciplinary teams.

The new situation makes it necessary to extend the competence, and thus the "professional language", in many professions by well considered and integrated concepts from informatics. From the point of view of programmers these are good news, since challenging and interesting cooperative tasks are ahead.

[Recent work relating to "profession-oriented languages" are found in [Bjerknes, G., et al. 1985], [Kaasbøll, J., 1986], [Nygaard, K., 1984], [Sannes, I., 1985] and [Sinding-Larsen, H., 1986].]

REFERENCES

Andersson, J., 1974: "Industriföretagets produktionsstyring" ("The Production Control of Industrial Enterprises"), Kungliga Tekniska Högskolan, Stockholm, Sweden, 1974.

Bjerknes, G., and Bratteteig, T., 1984: "The Application Perspective - Another Way of Conceiving System Development and EDP-Based Systems". Report from the 7th Scandinavian Research Seminar on Systems Engineering., Helsinki, 1984.

Bjerknes, G., Bratteteig, T., Kaasbøll, J., Nygaard, K., Sannes, I., Sinding-Larsen, H., and Thingelstad, G. 1985: "Gjensidig læring", ("Mutual Learning"), Report from Phase 1 of the "Florence Project on the Use of EDP in Nursing". Institute of Informatics, University of Oslo, 1985.

Boehm, B.W., 1981:"Software Engineering Economics", Prentice-Hall 1981.

Bratteteig, T., 1983: "Kommunikasjon i systemutvikling". ("Communication in System Development. Relevant Issues from the Social Sciences. Characteristics of Methods and Relevant Communications Situations".) Master Thesis, Institute of Informatics, University of Oslo, Oslo 1983.

Bråten, S., 1973: "Model Monopoly and Communication", Acta Sociologica, Vol. 16, 1973, No. 2, pp 98-107.

Bråten, S., 1983: "Dialogens vilkår i datasamfunnet", (The Conitions for Dialogue in the Computerized Society") Oslo University Press, Oslo, Norway, 1983.

Ciborra, C., 1981: "Information Systems and Transaction Architecture". International Journal of Policy Analysis and Information Systems, Vol. 5, No. 4, 1981.

Clocksin, W:F:, and Mellish, C.S., 1981: "Programming in Prolog". Springer-Verlag, Berlin-Heidelberg-New York, 1981.

Dahl, O.-J. and Nygaard, K., 1965: "SIMULA - a Language for Programming and Description of Discrete Event Systems". Norwegian Computing Center, Oslo 1965.

Dahl, O.-J., Myhrhaug, B. and Nygaard, K., 1968: "SIMULA 67 Common Base Language". Norwegian Computing Center, Publ. S-2, Oslo, 1968.

Ehn, P., and Kyng, M., 1985: "A Tool Perspective on Design of Interactive Computer Support for Skilled Workers". Report from the 7th Scandinavian Research Seminar on Systems Engineering. Helsinki 1984.

Fjellheim, R., Håndlykken, P. and Nygaard, K., 1974: "Report from a Seminar on System Description at "Skogen", Røros". Norwegian Computing Center, Publication S-65. Oslo 1974.

Fossum, E., editor, 1983: "Computerization of Working Life". Ellis Horwood Limited,Chichester 1983. (Distributed by Halsted Press, a division of John Wiley & Sons.)

Goldberg, A. and Robson, D., 1983: "Smalltalk 1980. The Language and its Implementation." Addison-Wesley 1983.

Holbæk-Hanssen, E., Håndlykken, P. and Nygaard, K., 1975: "System Description and the DELTA Language". Norwegian Computing Center, Publ. 523, Oslo 1975.

Håndlykken, P. and Nygaard, K., 1981: "The DELTA System Description Language - Motivation, Main Concepts and Experience from Use". In "Software Engineering Environments", edited by Hünke, H. Proceedings of the Symposium held in Lahnstein, Germany, June 16-20, 1980. North-Holland, Amsterdam, 1981.

Kristensen, B. B., Madsen, O. L. and Nygaard, K., 1983a: "Syntax Directed Program Modularization". In "Interactive Computing Systems" (Ed. Degano, P. and Sandewall, E.), North-Holland 1983.

Kristensen, B. B., Madsen, O. L., Møller-Pedersen, B. and Nygaard, K., 1983b: "Abstraction Mechanisms in the BETA Programming Language". Proceedings of The Tenth ACM Symposium on Principles of Programming Languages, January 24-26 1983, Austin, Texas.

Kristensen, B. B., Madsen, O. L., Møller-Pedersen, B. and Nygaard,K., 1985: "Multisequential Execution in the BETA Programming Language." Sigplan Notices, Vol. 20, No. 4, April 1985.

Kuhn, T. S., 1970: "The Structure of Scientific Revolutions". 2nd edition, The University of Chicago Press, Chicago 1970.

Kaasbøll, J., 1985: "Iformation Processes versus Data Processes", "Precedings" for the "Working Conference on Development and Use of Computer-Based Systems and Tools", Aarhus University, Denmark, 1985.

Kaasbøll, J., 1986: "Intentional Development of Professional Language through Computerization. A Case Study and Some Theoretical Considerations." Proceedings of the IFIP Working Group 9.1 Conference on "System Design for Human Development and Productivity: Participation and Beyond", May 1986, Berlin, GDR.

Marty, J.B., Hearn, A.C., Griss, M.L., and Griss, C., 1978: "Standard Lisp Report". University of Utah, Symbolic Group, Report No. 60, Provo, Utah, 1978.

Mathiassen, L. 1981: "Systemudvikling og systemudviklingsmetode". ("System Development and System Development Method".) Dr. Scient. Thesis, University of Oslo. Published as DAIMI PB-136, Dept. of Computer Science, Aarhus University, 1981.

Nossum, R., 1984: "Decision Algorithms for Program Unification". Institute of Informatics, University of Oslo, 1984.

Nygaard, K. and Bergo, O.T.,1973: "Planlegging, styring og databehandling. Grunnbok for fagbevegelsen". ("Planning, Control and Data Processing. Basic Reader for Trade Unions".) Part I - Introduction. Tiden Norsk Forlag, Oslo 2.ed. 1973.

Nygaard, K.and Bergo, O. T., 1975: "The Trade Unions. New Users of Research". Personnel Rewiev, 4, 2, 1975.

Nygaard, K., 1975: "DELTA-prosjektet og dets tilknytning til problemene i system-utvikling" ("The DELTA Project and Its Relation to the Problems in System Development".) Proceedings of the Aarhus Conference January 1975 on Working Metods in System Development. Department og Computer Science, Aarhus University.

Nygaard, K., 1977: "The Iron and Metal Project. Trade Union Participation". Proceedings of the CREST Conference on Management Information Systems 1977. London, Cambridge University Press.

Nygaard, K.and Håndlykken, P. 1981: "The System Development Process - Its Setting, Some Problems and Needs for Methods". In "Software Engineering Environments", edited by Hünke, H. Proceedings of the Symposium held in Lahnstein, Germany, June 16-20, 1980. North-Holland, Amsterdam, 1981.

Nygaard, K. and Fjalestad, J., 1981: "Group Interests and Participation in Information System Development". In "Microelectronics, Productivity and Employment", OECD, Paris, 1981.

Nygaard, K., 1984: "Profession Oriented Languages", in Proceedings from the conference Medical Informatics Europe 84, Brussels 1984.

Nygaard, K. and Sørgaard, P., 1985: "The Perspective Concept in Informatics", "Preceedings" of the "Aarhus 1985 Working Conference on "Development and Use of Systems and Tools", Aarhus University, Denmark, 1985.

Nygaard,K., 1986: "Information and Perspective", Working paper, Institute of Informatics, University of Oslo, 1986

Olle, T.W., Sol, H.G., and Verreijn-Stuart, A.A., (editors), 1982: "Information System Design Methods", North-Holland 1982.

Parkin, A., 1980: "Systems Analysis". Edward Arnold, London 1980.

Sandberg, Å., editor, 1979: "Computers Dividing Man and Work", Swedish Center for Working Life (Arbetslivcentrum), Stockholm 1979.

Sannes, I., 1985: "Computers as Tools for Nurses", in Proceedings of The 2nd International Working Conference on Communication in Health Care (International Medical Informatics Association), Stockholm, June 1985.

Sinding-Larsen, H., 1986: "Helsepolitiske realiteter versus "egentlig sykepleie" ved innføring av EDB", ("Political Reality versus "Ideal Nursing" when Introducing EDP"), SYKEPLEIEN, June 1986, Oslo.

Webster, 1960: "Webster's New World Dictionary of the American Language", New York 1960.