ANALYSIS OF USER NEEDS

B. Langefors
Administrative Information Processing,
University of Stockholm
S-10691 Stockholm, Sweden

1. The Problem

The design of information systems, IS, presents challening problems to the computer specialists or IS designers. They are also aware of this and are eager to meet this challenge. Consequently computer people are anxious to have the clients or users specify their requirements for the system as soon as possible. They do not expect the clients to present their needs in very precise terms and, hence, they try to interprete the requirements into "computer language". In doing so the computer people formulate their conceptions of the requirements according to their own ideas of the solutions.

Unfortunately, the computer people are not aware of the fact that to specify the requirements for the IS is at least equally difficult as to design the IS. Indeed computer system specialists tend to ignore the fact that the IS is just a subsystem of a larger system which it is to serve. This ignorance is serious. To develop the specifications for the IS means to design the main structure of the System of which IS is but a component. And the System is not a data processing system, it is a living organization containing people and different subsystems only few of which are data systems. It follows that the System design task is not a task of computer specialists.

It is quite some time since the recognition of the analysis of user needs - and of the System needs - as an explicit project activity was beginning to attract the attention of some researchers. One and a half decade ago I wrote [Langefors 1963-1]:

> "...great risks for a system to grow up, that will process large masses of data which are not used for decisions - and yet ignore important data - ...
>
> What is sorely needed in this area is a systematic ... technique for establishing the real needs for information within an organization. Thus it has to define the information needed, its volume,

> the time intervals at which it is required, and that at which it is available, the data from which it can be produced, and the process - or alternative processes - needed for its production and the form for presentation of the results ...
>
> An analysis of an information system must therefore, in the first place, be hardware independent. It has to provide an abstract definition of the system itself."

Likewise, a decade ago, Ackoff pointed out the need for more careful analysis of the user needs and for systematic approaches [Ackoff 1967]. Ackoff also pointed out the problem of collecting and processing masses of data which are not used:

> "It seems to me they suffer more from an overabundance of irrelevant information."

Also Ackoff emphasized the fact that crucial questions may be asked by people who know the organization but do not know much about computers:

> "The recommendation was that the system be redesigned as quickly as possible...
>
> The questions asked of the system had been obvious and simple ones. Managers should have been able to ask them but - and this is the point - they felt themselves incompetent to do so. They would not have allowed a handoperated system to get so far out of their control."

Main Problems

Some Basic Problems

- Who are the users?
- Who does the analysis?
- Why are systems not made use of?
- Which system, what problem, opportunity, or need?
- How combine overview, comprehension, user meaningfulness with concreteness and precision in details, in design documentation?

User Needs

. Needs for better organization, management, job design
. Organizational change, learning & development
. Needs for information, information service, computeraided data handling
. Social and personal needs

Problems with the information

. Relevance of information, for whom?
. Who are able to share the same data?
. How make sure the data provide the right information?
. Information for Everybody - what design implications?
. Information administration and data administration?

Who are the users?

Often when we talk about the users we have in mind all those who are affected by the system in some way, without having a designer's interests in the system. This is, for instance, quite reasonable in cases where the main question is about dominance of the experts or when one is concerned with other user/designer controversies. But the people affected are of many distinct kinds and are affected in distinct ways, often in ways such that the term "user" appears quite inadequate. Therefore, to study "user needs" or effects of systems upon people it is important to identify either the distinct kinds of affected people or the distinct ways people are affected. The latter way has several advantages and can be pursued by listing distinct roles vis-ã-vis the information systems. Then any person or work group can be analyzed into what roles he or it plays and to what extent.

Some roles are

. System sponsors - people who initiate and finance systems projects, because they estimate this is in the interest of the organization, regardless of whether or not they will be serviced directly by the system. Corporate management and the concerned line manager are in this role but any employee may be too, depending on the power structure at work.

. Information consumers - people who are aided in their work, decision-making or operating activities, by being serviced with information.

. Information suppliers - those who have to provide information to the system.

. System operators - those who have to manipulate the equipment, for instance interact with the systems on terminals.

. Other employed people - who are affected by the functioning of the system, for example workers whose work is scheduled by the system.

. Other affected people - who are not employees of the organization owning the system, for instance citizens vis-ã-vis a computerized taxation system.

It is important to recognize that one person may simultaneously have several of these roles. Thus a manager may operate a terminal to feed in data for a decision model and receive results at the terminal. He is then an information consumer and supplier and is also, part time, a system operator. He may also be one of the system sponsors. It is also important that the needs of these distinct roles are distinct.

Normally an IS is developed for the purpose of supporting information consumers. People in the other roles are, typically, affected by the system not because they wanted to use the system - being no information consumers - but because it is required as part of their job in order that the system will work. It is natural, though unfortunate, that the needs of theese people have attracted less interest. It is unfortunate because if the system was not designed with the purpose of serving them in their job, their jobs may become most affected. However, more recently the needs of these people have begun to be taken seriously into consideration.

Because many people are affected in the distinct roles they have, the effects on all the non-information consumer roles are complex and not yet well understood. It is not only how the system affects these people but also how they will experience these effects in the long run that must be clarified. Nevertheless these problems do not seem to be essentially distinct from other job design problems and it seems likely that there will, fairly soon, be a reasonable knowledge among system architects - and among the people themselves - how to handle these user needs.

The needs of the information consumers, though they have been investigated more early, appear to be the most problematic ones and they are likely to challenge us as researchers still for a long time. Because they are also the central problems of any IS design, they will be given the main focus in this presentation. Also, remember that I am talking of roles, not of persons, and any user, or affected person, may perhaps improve his situation through becoming also an information consumer. If so he will also profit from any advances we make in determining the needs of information consumers.

Who does the analysis?

Traditionally, computer programmers have expected the users to specify their needs. When this did not work system analysts entered the scene. In this way we had two kinds of specialists: the computer specialists - programmers and computer system analysts on the one hand and management scientists on the other hand. Often the same person tried to be all these three experts. The specified needs became more a reflection of what these experts believed to know about users' needs, perhaps after quick interviewing, than they reflected the real needs. Somewhat later, around 1965, it became common to have user representatives as members of the project teams. They were supposed to know the user needs. It turned out that the "system analysts", though often recruited from the user lines and trained by the computer manufacturer, behaved more as a kind of computer system experts than as real analysts of user needs. Their documentation was not understood by the users. As a consequence, the user did not know what the system would be doing and be requiring until the system was implemented.

Nowadays the insight is spreading that to determine IS user needs - at least those of the information consumers - requires a search-learning process through which the users develop an understanding of what their real needs are and what new opportunities have become available to them. As a consequence, the users emerge as the key persons in the task of analysing user needs. Information analysts appear to have an important part in aiding the users in learning and analysing, in doing the documentation in a way understandable to the users, while efficient as a design input for the data system design stages to follow, and in doing feasibility estimates.

It should be obvious that the new way of doing the analysis of IS users needs, that we have found necessary, raises questions of new methods. It also becomes clear that the analysts should have a social science and infology grounding more than one of a computer science or data processing background.

Why are the systems not made use of?

A frustating experience, to managers and system designers alike has been that the systems are not used or only used in their most trivial aspects. Expected gains have not been attained. Cost have greatly exceeded estimates, various difficulties have been encountered. Why do people not use the systems? Are they not enough motivated? Is this because they did not participate realistically in the design? Are the wrong goals defined for the systems? Are the data - and the system documentation - not accessible and understandable? Is the system architecture, the software, the hardware, unsuitable? Which system, what problem, opportunity or need? Before system design is taken on, one must decide which system to address. How should the system's borderline be drawn. How integrated with the organization? What problems should be attacked? Perhaps no problem but a new opportunity? What are the important needs?

How combine overview and comprehension with concreteness and precision in details? The term system is associated with things that have many, interdependent components. Accordingly, overview over the whole is difficult and is one of the most typical problems of system design. It is also a foremost difficulty with analysing user needs. How do they mutually interact? By what methods can one reach overview and combine this with systematic detailed analysis?

User Needs

The service from an IS may be needed for several reasons. The above list of user roles gives an indication in this respect. Better information makes possible better decisions. Better information service allows new organization structures, more effectively coordinated/directed local autonomy and may make for more convenient office work - though sometimes the opposite occurs.

With the present, rapid changes in economic and social environment organizations need to change, learn, and develop. Information service might be designed to aid in those aspects.

Social and personal needs are also, potentially, served by an appropriately designed IS.

Problems With Information

Relevance of Information. This is a problem not merely of what is relevant from an obejctive or "scientific" point-of-view, though this is often the only kind of relevance mentioned. The infological (or conceptual) view on IS has taught us that information must be "subjectively relevant" in order to become used. For instance, user's cognitive styles need be reflected. How may we go about to satisfy the two relevance requirements simultaneously?
Who are able to share the same data? IS (and data bases) presuppose that distinct people may use the same data but how may data be subjectively relevant to distinct people? What determines "shareability" of data?
How make sure the data correctly represent the intended information, to the intended users? This is the central infological question.
Information for Everybody. Not only democratic demands call for making the information available to everybody, also organizational efficiency and creativity makes this desirable. What can be done in this direction?

If the information is made available to everybody then it becomes natural to consider "everybody's needs" in designing the IS. How might this be approached? What additional information content will become relevant?

Information administration. To determine shareable information and shareable data and to care for system integrity and security there must be a central IS administration. To handle this by a special management may hurt the subjective relevance. How can both needs be compromised?

It seems likely that local, decentralized systems will be common. They will contain local data that do not need to be standardized or centrally available. But some data are likely to have to satisfy corporate requirements and must be controlled by a central body. How may one delimit such requirements to such data only, for which they are justified?

Problems with User Involvement

User involvement is important and needs to be improved in all applied technology projects but in the connection with IS design this is especially significant. Thus to define information needs one must be able to specify the information to have. This specification must be stringent in order to provide a basis for data design and computer programming. The stringency easily leads to unintelligibility for the various users. This is one factor that hampers user involvement.

The information needs and the interrelations between them makes for an extremely complex information structure or abstract information system involving many elements and linkages. In order not to get lost one must introduce abstract, overall information system models so, in addition to stringency, the users also will have to cope with abstraction if they shall have a real influence upon the IS design.

Another problem is that the term 'user' refers to many distinct roles. In other words many distinct groups of people need to be involved and as they are affected in distinct ways this causes distinct problems all of which affects the analysis work and methodology.

2. Available Results

2.1 Project Stages and Problem Areas of IS-design

Projects have since long been structured into stages and phases such as

Feasibility study
Main Study
System Design
Implementation
Operations, Maintenance, and Assessment

but this does not give much guidance as to what-to-do-when, in IS design. What problems are brought up in the Feasibility Study? What structure and content has Main Study?
In a systematic study of the problem categories involved in IS design and their interrelations, it was found that some distinct problem areas (or method areas or topic areas) are always involved and they present distinct kinds of problems and call for distinct skills. This result offered improved insight into how to organize and staff IS projects and what to teach and do research on [Langefors 1963-1, 1969]. It provided knowledge of various classes or problems involved in every IS project and established a framework for developing of methods and doing research on the distinct problem areas. Groups of tasks which require similar background can be assigned to common project activities and to designers/analysts with the appropriate skill and experience.

It was found that because an IS has the purpose of serving as a component in a larger system, its Object System, OS, the specification of the IS cannot be produced directly but must result from an earlier stage in which the main (or top-level) "subsystems structure" of the OS is designed. Importantly the main OS design stage is not a data system design and, hence, not properly delegated to the data system department (which should therefore not be referred to as the "Systems Departments"). In this way a specific problem area or topic area Object Systems Analysis & Design (OSA) was identified. The reader may alternatively interprete OSA as "organization systems analysis" if he so prefers.

Only after some object system analysis and design activities have identified some object system functions can the information needed of these functions be determined. This can be done in computer-independent terms which has significant consequences for methods development and user participation. Thus emerged the area Information Analysis & Design (IAD).

The areas OSA and IAD constitute the infological realm of IS design problems. Following them, logically, are the areas within the datalogical realm. Below the problem areas are presented in a tableau. The datalogical realm, which is less interesting in this context is quite condensed in the tableau, Fig. 2.1.1.

One important characteristic of the framework of problem areas is that after each completed activity in one area more knowledge has been obtained which is relevant for the cost and feasibility estimates in previous areas. This should be exploited for re-assessment of previously set goals. In the tableau feed-back arrows are shown, to point out this.

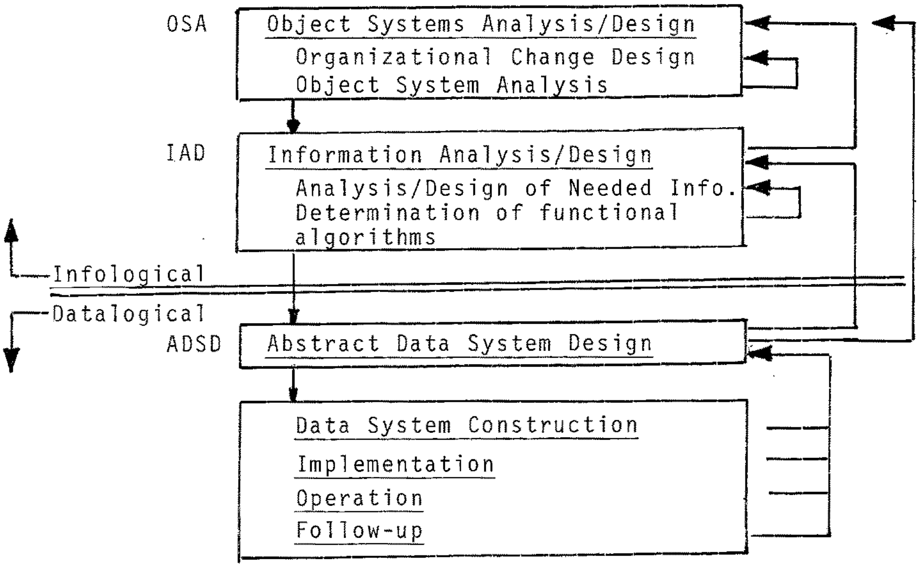
OSA
Object Systems Analysis/Design
Organizational Change Design
Object System Analysis
IAD
Information Analysis/Design
Analysis/Design of Needed Info.
Determination of functional algorithms
Infological
Datalogical
ADSD
Abstract Data System Design
Data System Construction
Implementation
Operation
Follow-up

Fig 2.1-1

Problem Areas & Project Stages

Advantages of these concepts

- Distinct classes of project tasks are recognized and problem areas are identified.
- More systematic research and methods development can be done
- Better knowledge of the skills required in distinct project stages and project activities was gained - henceforth neglected problems and methods needs got detected
- Important insights into how IS projects might be more efficiently structured and managed were obtained.

Especially

The areas Object System Analysis/Design OSA and Information Analysis/Design (IAD) became recognized as full-fledged work areas with their own needs for methods and skills, greatly distinct from computer science and data processing. Research in OSA and IAD got started and practical experience (for one decade, now) was acquired. These are the infological areas.

Thus the introduction of problem areas started the work on Analysis of User Needs.

The informatological (nowadays: infological) approach to IS design began to be developed, emphasizing the distinction between infological problems (OSA,IAD) concerned with people and their work and information use and the datalogical problems concerned with formal representation and computers.

Conclusions

Some object system design, change, and analysis should be done before information analysis, data design and programming (organization, people, jobs).

Change process, social interaction, learning, and identification of human values are crucial to the system design, as much as is technology.

The necessarily unstructured change processes must result in stringent design specifications that lead to control of the technical design - a clash between two "scientific cultures or paradigms". A challenge to human communication. Formal methods can only do part of the task.

As distinct social groups, with distinct values, will prefer distinct changes, the project problem arises as to whether there can be one single change project involving all groups or there should be distinct change projects which are then merged. Conflicting designs would then be developed and a negotiation process would follow.

Before it is determined what the real problem is, the boundaries for the actual system are undefined, hence the set of users is undefined.

Some problems:

- How to balance distinct user needs against each other in specifying design goals?
- How overview or comprehend the distinct needs and the effects of design decisions?
- Abstractions are necessary for overview, concrete details are needed for construction.

System design, being a social change process, may change power and will always be sensitive to conflict situations.

Problem Areas are not Identical with Project Stages

There is a strong, logical precedence order between the problem areas. Functions in the OS must be identified before their needs for information can be determined and this must be done before the data and programs can be designed that may satisfy these needs. But this does not mean that all of OSA must be done before any part of IAD can get started and so forth. For example OSA followed by IAD can be done on a crude level and then be succeeded by an OSA activity followed by a new IAD activity on the next finer level and so forth. Also, after concrete, detailed data system construction - for specific hardware - has been done, one may need to do some OSA work to modify the job design for some system operators. This will be done in order to adapt the job procedures to the operating characteristics of the computer system.

Other Work on IS Design Framework

A Framework For Information Systems Development which has many similarities to the above framework of problem areas was postulated in [McFarland,Nolan,Norton,1973]. It, likewise, differs from the traditional project frameworks by talking about the content of the distinct areas. The similarity with our framework is illuminated by the following quotation as well as the display of the framework, presented below.

> "With a subject as broad and encompassing as information systems, expediency requires that the whole be subdivided into manageable parts. If this subdivision is performed properly it becomes possible to address each part somewhat independently, for the dual purpose of understanding and generalizing. The overall framework is then used to integrate the parts into a meaningful whole."

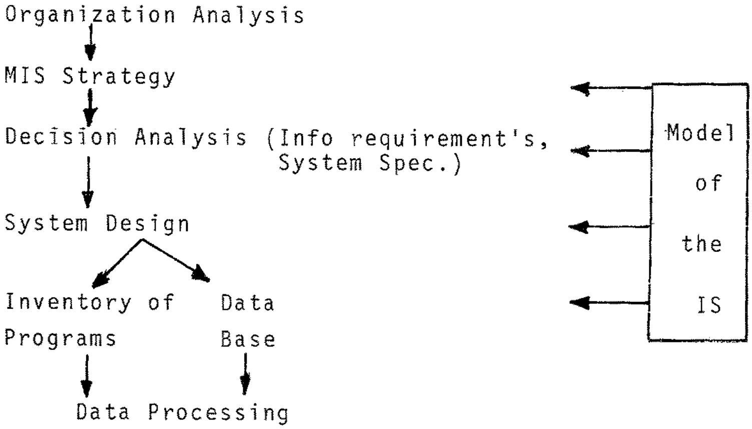

Fig. 2.1-2 The Framework of McFarland,Nolan, and Norton,1973

The similarity speaks for itself. I only want to point out that "System Design", as used here, had better be named "Data System Design".

It is of interest to notice that recent data base work also has begun to recognize distinct levels of the data base schema. Although this has a different background it is related to our problem area structuring and the framework by [Senko,1976] shows interesting similarity to ours. Senko talks of System Name-Based and User Name-Based levels and the Access Path Level, the Encoding Level and the Physical Device Level.

2.2 Infological and Socio-Technical Approaches to User Needs

The infological view is that because data are handled for the sole purpose of utilizing information in an organization and because many distinct data formats may be used to represent the same information it is necessary to describe information in a way which is accessible to the users and is independent of computer technology. The information specification states what is required or desired from the data processing. It ignores how the data system works, internally.

The infological view led to the introduction of specific, early analysis stages where the goals/desires of users are studied and identified, separated from the intricate technical construction work that will subsequently implement the desires. As a consequence, the infological approach [Langefors 1963,1966, Lundeberg 1968,1970, Nissen 1972, Sundgren 1973] subordinated the technological construction to social and human factors and developed methods and tools for making this possible.

The recent development of the database management has begun to address infological aspects for instance in defining information in terms of abstract relations [Codd 1970, Senko 1976] and developing query languages that allow the specification of the information to be retrieved in infological, datastructure-independent terms. Often, user needs may be defined in terms of such query statements.

The Socio-Technical Approach

The infological approach is, clearly, a sociotechnical approach. It was developed from a computer science and systems engineering background. Else sociotechnical approaches have been advocated and developed by distinct social scientists, for distinct kinds of systems. Mumford and Ward [Mumford and Ward,1968] presented a sociotechnical approach to data systems design. They propose that two distinct lists of design alternatives be worked out, one list of alternatives developed from the point-of-view of the users (the "system operators" mainly, in our terminology). The other list should be developed from a computer efficiency point-of-view. Then both lists are compared and a design alternative is selected which is as good as possible from both points-of-view. This scheme, like other sociotechnical proposals by social scientists, does not employ the strict subdivision into stages. This is natural because the social scientists were not aware of this kind of development in the IS engineering area. The sociotechnical approach, by putting the social and the technological drafting at equal levels implies a compromising between the human and the computer aspects. Contrary to this, the infological approach gives a priority to the human aspects, requiring from the datalogical designers to satisfy (realizable) infological specifications (or to feed back realizability problem information). It should be noted that the sociotechnical model described here was published a decade ago and that development is going on in the area. The Socio-Technical approach was also discussed recently in [Boström and Heinen 1977]. See also my discussion letter to MIS Quarterly regarding that article [Langefors 1978].

Object Systems Analysis and Social Change Design

The infological approach started by taking information definition as a basis for data and program construction. As a consequence the unavoidable, subjective human conditions for interpreting data and utilizing information came out as a fundamental analysis/design basis. In this way the problem area of Object Systems Analysis/Design emerged. The people, the stations, the functions in the OS (the organization) to be serviced, were analyzed as to what was in existence and what was desired for the future [Langefors 1963-1]. It was proposed by

Nissen [Nissen 1972] that the techniques for precedence and component analysis developed in the information analysis area ought to be applied also in OSA, after suitable amendments offered by Nissen. This turned out to be a fruitful approach and has come to be a standard part of infological systemeering (IS analysis/design). Its advantages are illuminated in [Lundeberg 1976].

An important development in OSA has been the study of IS development as a social change process by the ISAC group in Stockholm [Lundeberg 1977] and by Høyer in Oslo [Høyer 1975]. This work has made it increasingly clear that the OSA - and, as a consequence the overall System Design task - must be arranged so as to begin with a social change and learning process involving the future users as the main actors. In view of these findings (which are now accumulating) "analysis of user needs" is, fundamentally, a social learning and creating process and it is my impression that this should be integrated as a central area preceding all systemeering (or IS design) projects. With this insight it is no longer reasonable to go on as if one would be believing that the users know their needs or - still worse - as if computer experts or management science experts would know. The practical experience associated with this insight is still meagre, but what has been attained is quite convincing.

It is important to point out that the insights from our work on the early stage of systemeering as a social change process clarify that both the unstructered social processes and the well-structured integration and documentation of their results are necessary and worthy of being developed further. Thus the bridging of the two emerges as the central infological systemeering issue.

2.3 Structured Information Analysis

The infological perspective made it clear that data and algorithms are to be determined based on the needs for information - knowledge - provided to people through data. At the basis is therefore not scientific or logical laws but intuitive, non-logical user view of their work and its needs. No unique solutions can be expected and purely formal methods are inapplicable. But unaided intuition gets lost in complex systems problems. The best that can be hoped for is a structured approach which identifies semiseparate subdesign tasks and achieves an integration between them. Crude pictures alone permit total views but decisions regarding the whole thus comprehended must be formulated as a precise framework which strictly controls subsequent, detail design activities. Thus the latter essentially fills in detail content into the meshes of the framework. A structured design approach is thus obtained. A structured infological systemeering approach has been developed based on a few principles

1 The information product users want from the system determines the design task (together with goal information users want to prescribe).

2 Information precedence analysis, step by step, starting at user determined information products determines where to collect the information or from which other information to derive it and identifies the derivation processes.

3 The internal design of the information derivation processes can be done separately from the information precedence analysis.

4 To gain overview, crude pro-concepts of information may be identified first and then component analysis can be performed to identify finer pro-concepts, components, until e-concept (elementary information kinds) and e-messages (elementary messages) are identified. Precedence analysis can be performed on any pro-concept level, as deemed suitable, as well as on the elementary level.

5. Through the successive refinement of the IS structure which is obtained during the structured design, successively more is specified about the information processes and about the data but at the elementary level, process design (algorithm design) is completed.

The steps 1,2, and 3 were developed in [Langefors 1963-1 and 1966]. Steps 4 and 5 were formulated in [Langefors 1969] and further developed in e.g. [Lundeberg 1969] and [Langefors 1973].

The structured information analysis (and object systems analysis) based on the precedence and component analysis allows a precise guidance and a precise documentation to support any intuitive, social learning analysis/design process. The precision is obtained without using any computer language which would have hampered the understanding by various users. It is stringent enough to permit performance calculations on design alternatives, employ computer aid to documentation and data and program design and form a systematic basis for subsequent design tasks. The structure obtained can directly be represented through system matrices, decision tables and list structures but graphical representation is usually preferred by the users.

Structured IS analysis/design "automatically" produces crudelevel structured programming specifications so that a direct continuation to program design is available. The same is true for data base design.

A very brief and simplified illustration of a structured information analysis work is given in the "information graphs" Fig 2.3-1 and 2.3-2. It is assumed that the relevant users have decided to look into a prospect of an IS for goal steering of the sales activities. It is emphasized by the method that the organization should be careful to select "all" relevant people to participate in the study. Information analysts aid them in getting into à creative, social interaction and in doing the documentation. First one decides, we assume for the illustration, that "Operating Sale's Goals (g,t)" are to be proposed by the IS. This is represented by a rhomboid (representing an information entity), which is placed below (for output) the large box, "O", symbolizing the IS to be explored. The indication (g,t) is used to suggest that the goals will be set for a series of product groups "g" and a time period "t". Operating Sale's Goal (g,t) is a <u>pro-concept</u>, that is, an information kind which is undefined, so far, except for its name and the meaning it suggests. It is still concrete enough to allow the analysis/design team to do a first step of precedence analysis. This they do by raising, and debating, the question of what one would need to know in order to determine realistic and efficient Sale's goals, that is, what information precedents are relevant. The question of <u>how</u> to derive the goals from its precedents is postponed. The team starts by listing a large number of candidate precedents and after lenghty discussions, analysis and estimating they decide to use only two precedents, the (pro-concepts) Sales Forecast (g,t) and Other Corporate Goals. The latter is assumed to be provided to the IS from outside which is indicated by placing it above the system box "0" (zero). "0" is used to tag this box in order to indicate that it represents the whole of the IS being analyzed. "Sales Forecast (g,t)" is placed inside the box to symbolize that it will be defined within this system. The line "3" is part of the bundle of lines symbolizing the precedence relations. It will later be associated with a subsystem "3" intended for producing Operating Sales' Goals (g,t) from its precedents. The design of "3" is postponed.

The rest of the graph illustrates the remaining precedence analysis for the IS (System "0"). The graph represents the whole IS as is directly seen by the team from the fact that all information to be produced by the IS has been provided with all its specified precedents and the initial precedents (above the box) are all assumed to be available from outside. The team now makes a careful "workability" check on the system as specified, before going on to a more detailed analysis.

Note that at the present, crude (pro-concept) level it would not be possible to define the algorithms involved so this analysis/design was made possible by separating and postponing the internal design of the process from the IS structure design.

A more detailed analysis/design can be built systematically on the top graph "0" by addressing, one by one, the subsystems associated with the line "1", "2", and "3". This, again, requires the team and (possibly other teams) to engage in creative work but one is guided by having the precise tasks of starting with component analysis of the information product of one subsystem at a time and then repeat the precedence analysis on all the components. This work is illustrated in Fig. 2.3-2, for the subsystem 3. It can be seen that the team decided to define three components for the sale's goal. They are shown as subboxes within Operating Sale's Goal. They happen to be e-concepts, elementary information each occurrence of

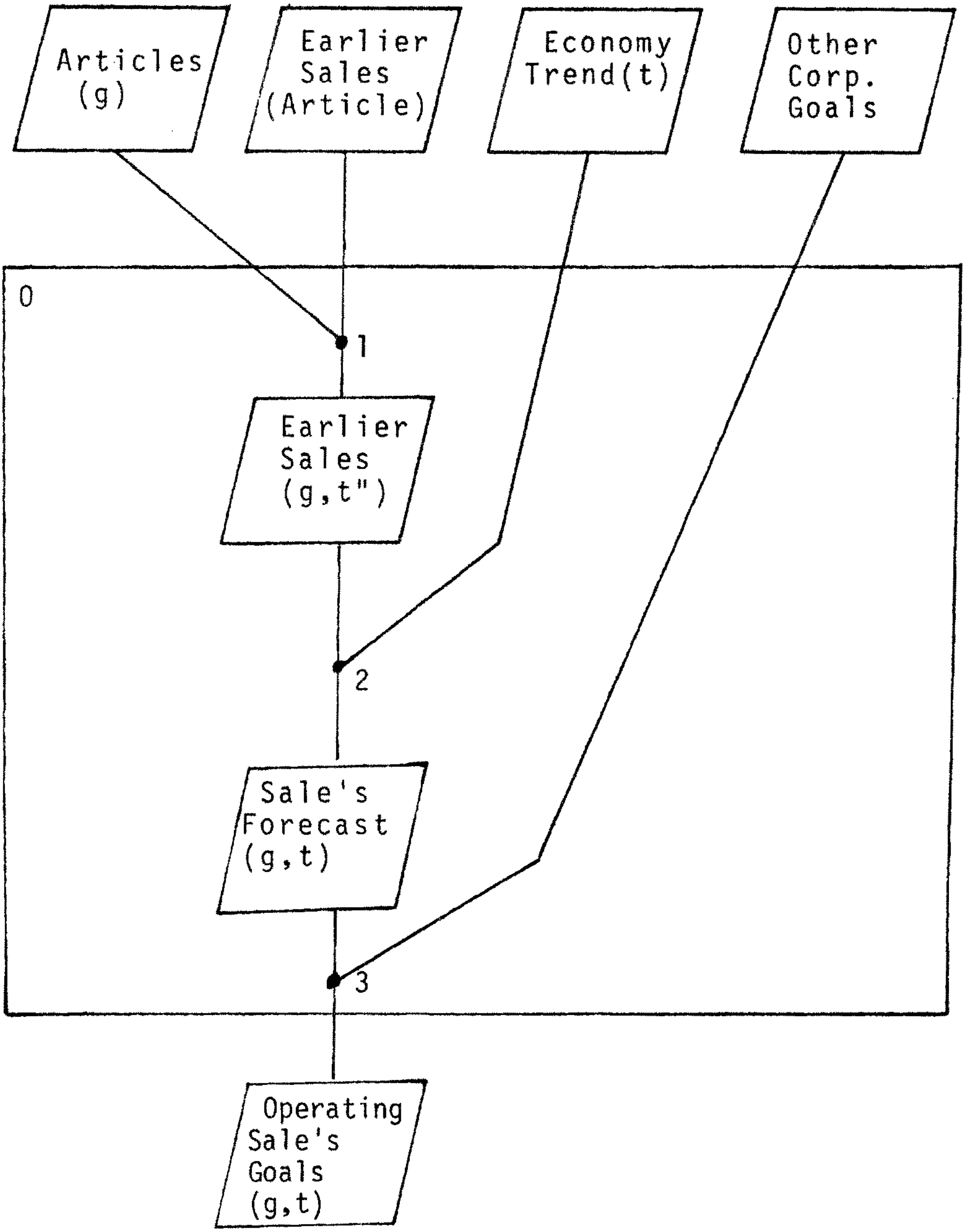
Articles (g)
Earlier Sales (Article)
Economy Trend(t)
Other Corp. Goals
0
1
Earlier Sales (g,t")
2
Sale's Forecast (g,t)
3
Operating Sale's Goals (g,t)

Fig. 2.3-1

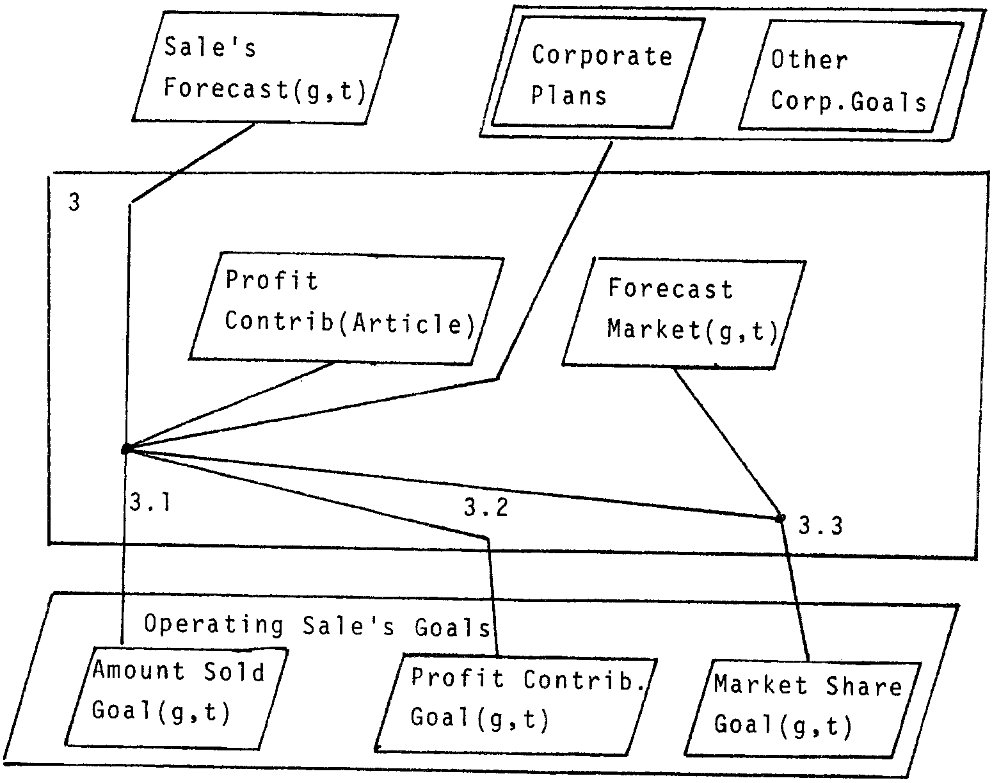
Sale's Forecast(g,t)
Corporate Plans
Other Corp.Goals
3
Profit Contrib(Article)
Forecast Market(g,t)
3.1
3.2
3.3
Operating Sale's Goals
Amount Sold Goal(g,t)
Profit Contrib. Goal(g,t)
Market Share Goal(g,t)

Fig. 2.3-2

which is one elementary message possessing one value for each value set of "g" and "t".

The Structured Method is Always Partial

It should be clear that the structured information analysis, as illustrated, can be done in any kind of organization and leaves it open to be separately decided whether more or less democratic design decision-making should be applied. This is to say, structured methods are like a network of roads. They make effective driving possible. But they do not drive. They can support many distinct kinds of driving, many driving methods and policies. Similarly, structured methods of analysis design help design teams organize the work and integrate the results. But they leave open the question of who participate in the design team or how to do the details of work. Methods need be developed for this as a supplement to the structuring methods.

In the case of information graphs, or other structure descriptions the details left aside, for separate documenting are e.g. quantitive data such as number and size of message instances of an e-concept, time, frequency and control conditions for the processing, semantic material for the e-concepts and algorithmic or mathematical details.

The fact that details may be represented separately, and piece by piece, implies that the way they are represented and documented is not very critical - as long as there are clear references to points in the structure description.

The structure documentation must be abstract in order to provide overview and comprehension. It may therefore be desirable to add concrete material such as concrete layout of reports or terminal screen pictures to show how specified information (messages) may be communicated between people and system.

The structured documentation has also the advantage of lending itself to computerized documentation and computer aids to analysis and design.

2.4 Other Structured Analysis Approaches

Some of the published work on the structured systemeering approach described in Section 2.3 in addition to the reference given above are [Lundberg 1976,1977], [Nissen T972,1976].

The structuring in the approach discussed above was characterized by stringent linking of unstructured chunks of work and by providing systematic linkage both horizontally (precedence analysis) on the same level of detail and vertically (component analysis) from cruder to finer levels. Postponing of local design tasks was also a key feature. Other approaches which attempt at systematic work provide, of course, some structuring but not any integration of all the mentioned characteristics. Also most structured approach proposals are emphasizing the program design efficiency whereas the above work is giving at least the same emphasis on identifying user needs.

One structured approach which has some similarities to the above, though not taking up all the mentioned aspects is HIPO (Hierarchy plus Input-Processing-Output) [Stay 1976]. It is more program-focussed.

One can see from the name already that HIPO also defines vertical and horizontal relationship ("hierarchy", Input-Output). One can also see that it does not postpone process definition and that it has not used the abstracting from "process-input-output" to information precedence relations. On these grounds it appears less developed conceptually. Also HIPO does not employ the strict distinction between data and information that has been found so important for the obtaining of a clear-cut structuring of the design task.

Some quotes from Stay's article will have to suffice here, to give an idea of what HIPO is or attempts.

> "The HIPO design process is an iterative top-down activity in which it is essential that the hierarchy chart and the input-process-output chart be developed concurrently, so as to create a functional break-down."

". User understanding and agreement on functional content are made easier.

. Missing or inconsistent information is identified early.

. Functions are discrete and are therefore more easily documented and, if necessary, modified."

The quotes illustrates the (partial) similarities in approach and itentions. It appears that development along the lines described earlier is beginning in more and more places.

2.5 Other Information Analysis Approaches

The infological, structured approach described earlier did not assume that users - or specialists - do know their needs. The approach was directed to guide in defining and documenting known needs but also - most importantly - to perform creative work in finding out unknown needs. Most other approaches presume that users do know the information (or the data) that they need.

Some of the more developed other analysis approaches are

- Data Analysis, Looking at existing data flows
- Decision Analysis, Looking at decision processes that need to be done
- Systematics, as developed by Grindley

An important, different approach is the experimentation with quickly implemented prototype systems.

Data Analysis and Decision Analysis appear as special and limited cases of information precedence analysis while ignoring the vertical (component analysis) aspects entirely.

Data Analysis [Munro & Davis 1977] ascertains information requirements by examining all reports, files, and other information sources. Managers are asked to state any additional information requirements. Data for which there is no perceived need are eliminated. As compared with structured, infological analysis this appears as a very limited special case. The organizational learning aspects, the vertical structuring and the formalized documentation aspects seem to be missing.

Decision Analysis [Ackoff 1967] [Mc Farlan,Nolan,Norton 1973], similarly to the infological precedence analysis, takes the needs for information and decisions as the starting points for the analysis. Contrary to precedence analysis it studies the decision process in its detail steps in order to identify the information required. Likewise contrary to the infological approach it restricts itself to the needs of managers. Because in the infological work it was detected early [Langefors 1963-1] that the information needs are not determined by procedure steps but by dependencies in reality, the Decision Analysis model reminds on the very earliest infological models (before 1963). Again, no formally structured tools appear to be developed in the Decision Analysis.

Systematics, as developed in [Grindley 1966] has several similarities with the infological analysis/design approach. Thus to obtain a precise definition of the information the concept ȯf "element" is introduced, corresponding to e-concepts and e-processes in the infological approach. But the methodological separation between the information and the processes is not employed in Systematics. There also is no counterpart to the pro-concepts and the component analysis. Systematics, like the infological approach, develops semi-formal description techniques. Decision tables are used as a central description tool.

One of Grindleys starting points was an observation from a Diebold study, around 1964, in which it was found that programmers spent less than half of their time writing and testing programs. The rest was spent on various System Queries such as

1. Changes made by the analysts to original specifications
2. Misunderstandings between analysts and programmers
3. Errors in specifications
4. Omissions in specifications

Many of the propositions in Systematics are in complete agreement with fundamental infology. For instance the proposition that two essentially distinct tasks go into any IS design work:

I Consideration of user needs
II Consideration of computer needs.

Experimentation with Prototype Systems

A different way to determine user needs is to directly implement extremely simplified prototype systems. Through ignoring performance problems and employing extremely simple design one can quickly implement application systems, using miniature data banks. One can thus allow the users to experiment with the system and then modify and extend the system successively, according to user demands. Afterwards, when the users feel satisfied one can redesign to achieve acceptable performance. In some application performance is not important so the prototype can be used operationally. This way of developing IS is sometimes advocated as an alternative to analysis. It seems to me that it is more appropriate to regard this approach as another way of doing the analysis. It is essentially similar to the infological approach in some fundamental aspects, such as involving users actively and realistically in the specification work.

The idea to use simplified fast implementation for "incremental system implementation" was proposed by Bubenko and Källhammar in the CADIS project at the Royal Institute of Technology, Stockholm, as a part of a development of Computer-Aided Design of Information System [Bubenko,Langefors,Sölvberg,1971, article by Bubenko and Källhammar]. This kind of ideas have appeared in different places, for instance [Bally,Britten, and Wagner,1977]. One interesting, recent development of this kind has been done by Staffan Persson, Stockholm School of Economics. He is using a small minicomputer which he takes along when visiting companies in Scandinavia, by car or by plane. He offers to implement a prototype of any system in three days, sometimes. He programs on-line while querying the users. These can then immediately experiment with the system and find out desirable modifications. Typically, one finds that questions arise early which may need several weeks for decision-making. It is important to identify such problems before large-scale implementation has been finished. This happens, typically, also in infological analysis work where then no programming at all has had to be done.

It is my estimate that the structured infological analysis is superior in some aspects, where abstraction and comprehension is at issue while the prototype experiment is superior in making the operative situation of the user concrete and live. It seems that the two approaches supplement each other, which ought to be done research on.

2.6 Documentation and Analysis/Design Aids

All systematic or structured analysis/design methods need be supplemented with some documentation/description methods. In the IS analysis/design area some one-dimensional languages have been proposed. Examples are the system algebra language (supplemented with Algol or COBOL statements) [Langefors 1966] and PROBLEM STATEMENT LANGUAGE (PSL) [Teichroew 1972]. PSL is, in the ISDOS project, supplemented with PSA, Problem Statement Analyser which checks the statements for consistency and some kinds of completeness. The 1-dimensional languages are poor system-languages in the sense of not aiding overview very well. Their advantage seems to be expected to be that they can be interpreted and analyzed by computers. But then one should not ignore the fact that 2-dimensional descriptions are also quite easy to input to a computer.

Two-dimensional languages are useful for systems descriptions because they employ both dimensions of the document paper to better aid overview, which is one of the central problems with complex systems. System matrices, precedence and incidence matrices were introduced for IS descriptions both for needs analysis and design computations, based on system algebra, in [Langefors 1963-1, and 1966]. These matrices describe the structure of the system whereas decision tables were available for system logic descriptions [Langefors 1970, 1970 version of 1966], [Grindley 1966].

As was observed by Jäderlund in 1974, decision tables may be treated as special parts of the IS incidence matrix so that a combined matrix, the Process Control Matrix, [Jäderlund 1976] was obtained which describes, concurrently, the IS structure and the process control logic, see also [Langefors and Sundgren 1974].

As we have already seen, formalized graph presentations have been developed and used extensively as documentation tools for user needs analysis, both information analysis and object system analysis (organizational analysis). This kind of presentation seems to be preferable to users though a systematic comparison with matrices has not been undertaken so far. It is often held against the use of graphs that they cannot be input to computers. Such statements, however, are built upon ignorance. It is rather easy to input formal graph information to computers as may be concluded from the facts that such graphs are isomorphic to incidence matrices. A little more difficult is to produce graphs by computers but reasonable solutions do exist.

2.7 Computer Aids to Needs Analysis

It was pointed out early [Langefors 1963-1,1966] that formalizing IS analysis/design concepts to improve understanding would also have the advantage of allowing the use of computers as an aid to IS design but also to the analysis of user needs. The possibility of computer aid to IS design has aroused a great deal of interest in many places. It is a question, though, whether we have not underestimated the other kinds of knowledge that we need to have before it will be worthwhile to use computer aided needs analysis. As always, it is easy to jump to formalizing and automating too much too early.

3. Some Practical Experiences

3.1 Early Experience

It has long been known that data processing systems often were failures and even those seen as reasonably successful were questionable in many respects. Costs and time for implementation were typically twice the estimated figures, or more, and gains were half. Users failed to use the systems, maintenance was extensive and endless. This sort of experience was published widely in the 1960's. As late as in the early 1970's one could still see reports of this kind by (still) surprised and disappointed authors. One such work [Dew & Gee 1972] studied the use made of budget information. They report for instance

> "in a study of four manufacturing enterprises with "well established budget control systems" 60 middle managers were queried. It was found that 32 ignored completely their budget information and 12 made only limited use of it."

3.2 Experiences with Infological Analysis Applications

Already in the early application work (mid 1960's) some of the intended advantages were clearly realized:

- users did find the documentation possible to understand and appreciated the absence of computer language, for instance, users have by themselves corrected mistakes in system graphs
- thanks to the precision of description users detected mistakes and asked for changes - typically 10 to 20 changes were done on the earliest graphs - and, thus, faulty implementation was avoided
- users learned about their own work through participating in the analysis - they stated they had reached an overview of their own work and its place in the organization, never possible before.

- it was easy to replace analysts, thanks to the structured documentation. This is important in all large projects.

Later on (in the 1970's) some "finer points", not visible to the researchers earlier, have come out:

- formal training should be provided to the users before project starting
- users should be encouraged to take control in the analysis work and do some of the documentation, themselves
- users, from many organizations, have become rather enthusiastic about the infological analysis approach, for instance, several of them have emphasized the importance that universities continue research and teaching in the area
- all kinds of users, not only managers, have taken part, for instance, in some cases the factory workers were able to specify requirements for economic information that would illustrate the "value added" of their work in the company profit
- in some cases it has been possible to mail the documentation to the computer programming team who then mailed the programs to user representatives who took care of the implementation.

As already mentioned, experience lead to expanding on the "social" part of the information analysis and object system analysis/design and organizational change:

- several projects falter early because of social conflicts between distinct user groups, a conclusion has been that a <u>social feasibility</u> study ought to be done before other feasibility studies. No project should be started unless the social environment is cooperative
- in research projects where the users took over most of the analysis work positive results have been most certain

- in projects where teams of users for doing the social change design have been established, operative valuable organizational innovations have been created and implemented in the system
- it has become clear that "information analysis" has to be a creative, learning process, not just a mapping of known needs
- users tend to gain power through working with the structured analysis; this may generate conflicts with other users who are brought into the project later on - hence it seems important to bring in all people from the beginning [Lundeberg 1977].

3.3 Some Observations from Empirical Research

Problem with User Representatives. Efforts at really involving users in IS design projects has been taken since long. It used to be found that this didn't work. We know now that it couldn't with traditional, technology dominated analysis methods. More recently user representatives have been more involved, in some cases. This seems to be more the case with on-line systems, especially with local, minicomputer projects. A disturbing finding, reported for instance by John Kjaer and by Peter Neergaard from distinct projects at the Copenhagen School of Economics, has been that if user representatives do get to function in design teams then they quickly get regarded as technological experts and, thus cease to be really representative. I have, myself, seen this happen, also, at different places.

One-Line Users - System Operators

Many of the systems studied in recent research have been on-line systems. There is a clear shift in findings in that whereas batch systems used to be problematic from both user aspects and business aspects, studies of on-line systems have regularly shown results where positive user reactions dominate the overall estimates. On the other hand it is also regularly found that people who work for long stretches of time have physical problems of various kinds caused by unsuitable design of terminals, desks and chairs. Job rotation, daily, has been found important for these reasons.

IS Influence upon Social Communication. The design of modern IS tends to replace man-to-man communication by man-to-computer terminal communication. This may reduce social contacts in a serious way which has been pointed to by some researchers. However, recent studies have also shown that IS may have the opposite effect in encouraging distinct system users to take up contacts with each other that they did not have before [John Kjaer,1977,Copenhagen School of Economics, personal communication].

4. Ongoing Work

The study of user needs, both as a separate research field and as a specific problem area in IS projects is quite recent, as we have seen, and not too much is going on or known to be. Indeed it is hoped the present conference will contribute to improving the situation in this respect. Only a few indications can be presented here.

Communication Aspects in IS design

The structured infological approach and its emphasis on user understanding, hierarchic descriptions and graphical aids is clearly oriented towards communication but that term is not emphasized in published work. Communication has many associations with IS design but three may deserve special attention: (1): the design documents as vehicles for communication between users, analysts, programmers, operators and maintenance staff, (2): the system design process as a communication process and (3): the use of data processing systems as communication systems.

Empirical research on description techniques from an interspecialist communication perspective was reported in [Willoughby and Arnold]. I hope work of this kind is going on and I think more of it would be desirable. Work on methods concerned with human communication to be included in IS design work is presently being pursued at the Royal Institute of Technology, Stockholm [Lundeberg 1977].

The hypothesis that it is useful for the study of existing IS to regard them as instruments for human communication was forwarded and pursued by Nissen [Nissen 1976]. It seems promising also from the needs analysis point-of-view. Nissen points out that the use of a formalized language restricts the subject that can be communicated. Computerization restricts further. It is important, when formalized channels are introduced, so as to replace informal communication, to determine the communication that becomes ruled out by the change and to introduce appropriate means of compensation.

Human Information Processing Aspects

The infological approach emphasizes that data must be designed with their use in human information processing in mind. This is the essence of distinguishing between data and information. This area is, of course, closely related to that of the human communications approach. Recently, much work has been started in this area. Ongoing work is indicated by some recent publications. Barkin, at the University of Utah, and Dickson at the University of Minnesota [Barkin and Dickson 1977] report on empirical work, on hypothesis testing, regarding information system utilization as dependent on the cognitive style of the decision-makers. Similar studies are reported in [Lucas 1975].

Dialectical Needs Analysis

We pointed out that structured methods partition the total analysis and design task into subtasks. The boundary conditions for subtasks and the linkage between them is handled precisely by the structured method while the content of each subtask is left entirely open by the structured method. Various methods and various decision power systems can use the same structuring methodology but lead to choice of distinct subtask methods. The use of creative learning processes within subtasks was mentioned. An alternative method to be used for subtask activities is "the Dialectical Method". This is thus not an alternative to structured methods but, rather, one possible supplementary method to be considered for some subtask activities.

Dialectical approaches to systems design have been advocated and worked on by Churchman and followers, such as Ian Mitroff [Mitroff, 1972] and, apparently, is still going on. They point out that information and epistemology are closely related, a position which is very similar to that of infology. Philosophers have discussed various systems of epistemology or inquiring systems. Different kinds of information are associated with each of them, Mitroff asserts. Churchman and Mitroff present a list of distinct systems of inquiry

Leibnitzian: purely formal, deductive

Lockean : experiential, inductive, consensual, empirical

Kantian : both formal and empirical, reconciles Leibnitzian and Lockean

Hegelian : conflict, synthetic design,the strongest possible conflict on any issue

Singerian : synthetic, interdisciplinary, attempt to integrate scientific, ethical, and aestetic modes of thought.

Compared with the infological framework the Leibnitzian corresponds to the datalogical, Lockean corresponds to the empirical, data collection aspect and the Kantian, Hegelian and Singerian correspond to various aspects of the infological concept of "interpreting structure" or frame of reference. Thus the Churchman and Mitroff list does not take us outside infology but, rather, provides additional content inside it. An example is the dialectical, Hegelian approach as suggested by Mitroff. On every important issue one should find two experts with opposing views (corresponding to possessing distinct data interpreting structures in the infological sense). They should be "equally creditable". These experts should perform a strongest possible debate on the issue and be watched by the decision-maker (the users we would say) who should create his own synthesis. Clearly this is one proposal for trying to handle the infological problem, mentioned earlier, of how to reconcile subjectively relevant and "objectively" relevant information specifications. Let me end this account of Mitroff's work with a quote from him [Mitroff 1972].

> "... the search for a common, consensual position may be ill-fated, if not outright dangerous. It may be more important to have a methodology that allows one to make sense out of direct confrontation than to eliminate confrontation by either suppressing it or whishing it away."

Empirical Research to Evaluate Needs Analysis Methods

When alternative analysis methods become available it becomes interesting to evaluate their respective advantages and disadvantages. This turns out to be an extremely complicated task. It appears to me that we do not yet know what criteria might be relevant for the comparison. Nevertheless we ought to get started. Hence the work taken up in this direction in [Munro and Davis 1977] is welcome beginning. Munro and Davis try to compare two variants of information needs analysis, the "data analysis" and the "decision analysis". They formulate a number of hypotheses and design an experiment to collect data for testing them.

REFERENCES

Ackoff, R.L., 1967, Management Misinformation Systems, Management Science, Vol. ü4, No 4, December 1967.

Bally, Britten, and Wagner, 1977, The Prototype Approach to Information Systems Design and Development, Information and Management, Vol.1, No 1, pp 21-26.

Barkin, S.R., and Dickson, G.W. 1977, An Investigation of Information System Utilization. Information and Management 1(1977) North Holland Publ.

Boström, R.P. and Heinen, J.S., MIS Problems and Failures: A Socio-Technical Approach, Part I MIS Quarterly, September 1977, Part II MIS Quarterly, December 1977.

Bubenko, J., Langefors, B., Sölvberg, A. (eds), 1971, Computer-Aided Analysis and Design, Studentlitteratur, Lund, Sweden.

Codd, E.F., 1970, A relational Model of Data for Large Shared Data Banks, CACM 1970.

Dew & Gee, 1973, Management Control and Information , MacMillan,London.

Grindley, K. 1966, Systematics - a Non-Programming Language for designing and Specifying Commercial Systems to Computers. The Computer Journal 9(2)Aug. 1966.

Grindley, K. 1975, Systematics, A new Approach to Systems Analysis, Mc Graw-Hill 1975.

Høyer, R., 1975, Systemeering and Social Reality: Formal and Informal Aspects of Administrative Control, In Lundeberg, M., Bubenko,J.(Eds), Systemeering 75, Studentlitteratur, Lund, Sweden.

Jäderlund, Chr.,1976, "Systematik" in Data/4-1976, pp 35-42.

Langefors, B., 1963-1, Some Approaches to the Theory of Information Systems BIT 3 (1963) 229-254, Copenhagen, Denmark.

Langefors, B., 1963-2, Toward Integration of Engineering Data Processing and Automatization of Design, In Vistas in Information Handling The Augmentation of Man's Intellect by Machine (P.Howerton,ed) Spartan Books.

Langefors, B., 1966, Theoretical Analysis of Information Systems (THAIS) Studentlitteratur, Lund, Sweden.

Langefors, B., 1969, Management Information System Design, IAG Quarterly 2 (No 4).

Langefors, B., 1970, Integrated Control by Information System, Effectiveness and Corporate Goals, in Management Informations Systeme, Grochla and Szyperski (eds) Betriebswirtschaftlicher Verlag, Dr Th. Gabler GmbH, Wiesbaden 1971.

Langefors, B., 1973, Control Structure and Formalized Information Analysis in an Organization in Grochla & Szyperski (eds) Information Systems and Organizational Structure, Walter de Gruyter, Berlin, New York 1975.

Langefors, B., 1978, Discussion of [Munro and David 1977], in MIS Quarterly, September 1977.

Langefors, B. and Sundgren, B. 1974, Information Systems Architecture, Mason/Charter, New York

Lucas, H.C., 1975, Why Information Systems Fail Columbia University Press, New York.

Lundeberg, M., 1976, Some Propositions Concerning Analysis and Design of Information Systems, TRITA-IBADB-4080, Royal Institute of Technology, Stockholm. Doctorate thesis.

Lundeberg, M., 1977, Utilization of New Information Systems Development Methods in Practice - Perspectives and Prospects, Information Processing 77, B.Gilchrist, Editor, IFIP, North-Holland Publ.

Mitroff, I.I., 1972, Dialectial Inquiring Systems: A New Methodology for Information Sciences, Journ. of the Am. Soc. for Information Science - Nov.-Dec. 1972.

Mumford, E., and Ward, T.B., 1968, Computers: Planning for People, London: B.T. Batsford

Munro, M.C. & Davis, G.B., 1977, Determining Management Information Needs ... MIS Quarterly/June 1977.

Nissen, H.-E., 1972, A Method for the Description of Object Systems in Information Systems Work, TRITA-IBADB-4403, Royal Institute of Technology, Stockholm.

Nissen, H.-E., 1976, On Interpreting Services Rendered by Specific Computer Applications, Liber/Allmänna Förlaget, Stockholm (Doctorate thesis) Royal Institute of Technology, Stockholm.

Senko, M.E., 1976, Specification of Stored Data Structures and Desired Output Results in DIAM II..., Lockeman and Neuhold (eds), Conf. on Systems for Large Data Bases, Brussels 1976.

Stay, J.F., 1976, HIPO an Integrated Program Design, IBM System Journal No 2 1976.

Teichroew, D., 1972, A Survey of Languages for Stating Requirements for Computer Based Information Systems, Proc. FjCC, p. 1203-1224.

Willoughby, T.C., and Arnold, D.C., Communication with Decision Tables, Flowcharts and Prose, (The copy I have does not have ref. to the publication. I believe it is the ACM publication Data Base. The authors were with the department of Accounting and Mgmt Inf.systems College of Bus.Adm., The Pennsylvania State University, University Park, Pennsylvania) 1971?