On the use of Semantic Models for specifying Information Needs

Erik Malmborg

R&D Report
Statistics Sweden
Research - Methods - Development
1989:5

INLEDNING

TILL

R & D report : research, methods, development / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1988-2004. – Nr. 1988:1-2004:2.
Häri ingår Abstracts : sammanfattningar av metodrapporter från SCB med egen numrering.

Föregångare:

Metodinformation : preliminär rapport från Statistiska centralbyrån. – Stockholm : Statistiska centralbyrån. – 1984-1986. – Nr 1984:1-1986:8.

U/ADB / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1986-1987. – Nr E24-E26

R & D report : research, methods, development, U/STM / Statistics Sweden. – Stockholm : Statistiska centralbyrån, 1987. – Nr 29-41.

Efterföljare:

Research and development : methodology reports from Statistics Sweden. – Stockholm : Statistiska centralbyrån. – 2006-. – Nr 2006:1-.

R & D Report 1989:5. On the use of semantic models for specifying information needs / Erik Malmborg.
Digitaliserad av Statistiska centralbyrån (SCB) 2016.

urn:nbn:se:scb-1989-X101OP8905

On the use of Semantic Models for specifying Information Needs

Erik Malmborg

R&D Report
Statistics Sweden
Research - Methods - Development
1989:5

ABSTRACT

The thesis-work summarized in this paper is based on results from two different projects I have worked with at Statistics Sweden. The title "On the use of Semantic Models for specifying Information Needs" indicates a common theme of both projects.

The largest part of the paper is additional material from my work on the further development of the Statistics Sweden approach to system development. The earlier papers from this work (Malmborg-82, -83, -84) are some years old, and there is a need to add some new results from the work. The main additions are a framework for discussing perspectives on system development and a presentation of the proposed extensions to the SCB model for systems development.

The latter part of the paper is introductory material on statistical databases. This is intended to be background material for the reader of Malmborg -86, -88. These papers are results from a project on semantic models for statistical data bases.

The reports constituting this thesis (Malmborg-82, -83, -84, -86, -88) are summarized in an enclosure.

ACKNOWLEDGEMENTS

Most of the work reported in this thesis has been performed in the course of my projects at Statistics Sweden. The main motivation for these projects naturally is the needs of Statistics Sweden. The possibility to use parts of the work for this thesis is mainly due to the influence of my manager and advisor Bo Sundgren. It is his influence that has made it possible to give the projects an "academic" orientation. Other managers at Statistics Sweden who have supported the work are Christer Arvas, Lars Olsson and Staffan Wahlström.

For valuable discussions in different contexts I want to thank more people than I can remember at the moment. Here some of the most influential are named.

At Statistics Sweden I specially want to mention Bo Sundgren, Björn Nilsson, Sven Felme and Birgitta Lagerlöf.

At Assistor AB I specially want to mention Roman Zielinski and Olof Risberg.

At Stockholm University I specially want to mention Börje Langefors and Janis Bubenko, who both have given valuable comments at seminars where my work has been discussed.

CONTENTS

1 Introduction
1.1 Background
1.2 Structure of this thesis-overview

2 Semantic data models

3 Semantic models as a base for system development
3.1 Different perspectives on system development
3.2 The further development of the SCB model
3.3 Remarks on the transformation from an OPREM-P type model to an OPREM-D type model
3.4 On the needs for CASE-tools supporting OPREM-P and OPREM-D

4. An introduction to Statistical and Scientific Data Base Management

1. Introduction

1.1 Background

This thesis-overview presents a summary of certain results from two methods oriented projects I have worked with at Statistics Sweden. The first project was on system development. More specifically the aim was to extend the Statistics Sweden approach for system development. The second project is on statistical databases. This project has only been partly finished.

In the course of the two projects I have published several papers. Five of these are relevant to the theme of this thesis. Enc. 1 contains summaries of these papers. The summaries are from the perspective of the thesis, and are complementary to the abstracts in the reports. All five papers have been written for scientific conferences, and the last four were accepted and published in conference proceedings. Each conference series has a topic and a common frame of reference. As the size of conference papers is limited by the rules for submission, this common frame of reference is often not explicit in the papers. In this overview I will try to give some of the "missing" background that can help the reader who is not specifically aquainted with the topic of e.g. statistical and scientific databases.

1.2 Structure of this thesis-overview.

The two projects mentioned above both are oriented towards aspects of semantic models.

The first project takes a broad view and tries to apply semantical modelling on most parts of system development, i.e. not only on the data base aspects. The papers from this project are:

Malmborg-82: The OPREM-approach, An extension of an OPR-approach to include dynamics and classification. (S/SYS-E12)

Malmborg-83: An analysis of systems design methodologies using the ISO-framework. (S/SYS-E14)

Malmborg-84: Stepwise formalization of information systems specifications by extending a simple object-oriented approach. (P/ADB-E20)

The title of Malmborg-82 is somewhat misleading. "classification" should be substituted by "inheritance" or "generics".

The second project is much narrower and concentrates on semantic models for statistical databases. The papers from this project are:

Malmborg-86: On the semantics of aggregated data (U/ADB-E25)

Malmborg-88: Design of the user-interface for an object-oriented statistical data-base (R&D-report 88:11)

The present overview contains 3 sections in addition to this introduction.

Section 2 is a discussion on some aspects of semantic data base models. Only papers with a clear impact on my work are discussed. No attempt to give a comparative overview of conceptual models for databases is made.

Section 3.1 is on different perspectives on system development. A general model for the interaction of different aspects is presented.

In section 3.2 the proposed extensions to the Statistics Sweden approach are presented. The approach is named OPREM-D (D for data). Major parts of the work presented in the papers Malmborg-82, -83, -84 is on an earlier version that I today call OPREM-P (P for process).

Under the development of OPREM-D, some different ways for findig a sound mechanical transformation between

OPREM-P models and the corresponding OPREM-D models have been investigated. This has not been successful, but the aims and attempts are described in section 3.3.

The final, part of section 3 (3.4) is devoted to the need for computer support for OPREM-P and OPREM-D. This is what today is called CASE-tools. The partial tools implemented are described.

Section 4 is mainly background material on statistical databases. This is hopefully of value for the reader without prior knowledge of the area.

There is no special section on "The use of semantic models for specifying information needs", but this is thought to be a main theme of my work in both projects. I have tried to emphasize these aspects where appropriate.

2. Semantic data models

The concept of "semantic data model" denotes data models designed to provide richer, more expressive concepts with which to capture more meaning than is possible when using "classical" data models. The classical data models are the hierarchical, network and the relational models. The term "semantic data model" is used with this meaning in e.g. Brodie-84 and Peckham-88. The latter paper is published in ACM Computing Surveys which indicates that the concept is established.

Well-known representative semantic data models are Entity-relationship model (Chen-76), RM/T (Codd-79) and Daplex (Shipman-81). There exist at least a hundred more models published in the literature, but those just mentioned are of special interest for this thesis. Entity-Relationship models and RM/T have been used as a basis for semantic models of statistical data bases (cf. Malmborg -86,-88).

Overviews of semantic data base models can be found in e.g. Kerschberg-76, Lindencrona-79, Tsichritzis-82, ISO-82, Brodie-84, Peckham-88.

Languages for retrieving data from databases based on semantic models can be found in several papers e.g. Shipman-81, Wong-82 and Sundgren-88.

Wong-82 has had a special impact on the work described in Malmborg -86, -88. The "language" described in Wong-82 is more of a graphical user-interface, than a traditional language. By navigating in a graphically displayed EAR-type model and manipulating it, the information need is specified. An equivalent question in a more traditional relational database language is generated and evaluated. There are other similar systems (e.g HYBRIS by Björn Nilsson et. al. at the Swedish Institute for Systems Development ("SISU"), but the GUIDE-system in Wong-82 has been most influential on my work (as presented in Malmborg -86, -88)

3. Semantic models as a base for system development

3.1 Different perspectives on system development

In many approaches a model of the Information System (IS) is produced during the system development. Often it is proposed that this model should be in "the terms of the user" or that the information/data structures should be seen from "the view of the problem" or that we should build a "semantically oriented" model of the IS (or database).

I advocate a somewhat different approach. The semantic model of the IS should be obtained by firmly basing this model on a model of something else. In broad terms I advocate that the structure of the IS (programs and database) should be based on models of something outside the IS.

The sub-proposal that the structure of the database should be based on a model of something outside the IS can in simple cases be realized by an Object-Property-Relationship model (used in the SCB Approach) or a Chen-type EAR-model (e.g. ISO-82). Models of this type can be viewed as either a model of "reality" (or sometimes the "enterprise") or of a data base. In the latter case it is a "conceptual" model of the data base (ISO-82 et al). In simple cases the distinction is not important but in more complex situations it is. The data base can be directly based on the model(s) e.g. by having one file (or relational data base relation) for each object class. In this paper the term "semantic model" is used to include both "reality-models" and semantically oriented IS-models.

The sub-proposal that the program structure should be based on a model of something outside the IS is the core of JSD (Jackson Structured Development). In JSD the structures are derived by modelling the order of events in the Universe of Discourse (or "Reality"). With the "structure of the program" is meant the iterations (loops) and selections (if-clauses) that the structured program is composed of. Still more specifically the number of cycles (or turns) of each iteration should be based on the outside model. These requirements on the programs are much harder than the requirement that they should be "structured" in the traditional sence (Dijkstra-72 ,Mills-72).

In most situations we need, in addition to a (semantic) reality model, a model of the information needs. I think this aspect is important. However, it is rather well handled in established approaches like ISAC (Goldkuhl-80 and other books) and MBI/SAK (Hugosson-83, Wigander-79).

The aspects emphasized by those approaches will be called "pragmatical" in this report. Hugosson-83, which is representative, states that "the basis of information (modelling) should be its use, and possible effects of this, not the information itself". The approach in Hugosson-83 is named MBI (In Swedish: "Mål-Beslut-Information"). Roughly the system development process acording to MBI contains the following steps:

1. A funtionally oriented model of the enterprise is created.

2. Based on the goals (Sw. Mål) of the enterprise operative decision- (Sw. Beslut) points are identified.

3. Information (Sw. Information) is needed for the decisions. These information requirements are identified.

4. The necessary information systems are identified and described as a basis for implementation.

A problem with pure pragmatical approaches, is the lack of precision in the modelling. To specify the information needs they refer to objects or domains, but in the perspective of my analysis, this is not enough. A lot of information needed to implement the information system concern the inherent relationships between the value-sets of the domains.

In order to create an information system model on the required level of completeness it is thought to be a need for several types of models. Different approaches for system development concentrate on different aspects. The proposed relationships between the models are (c.f. Malmborg-84):

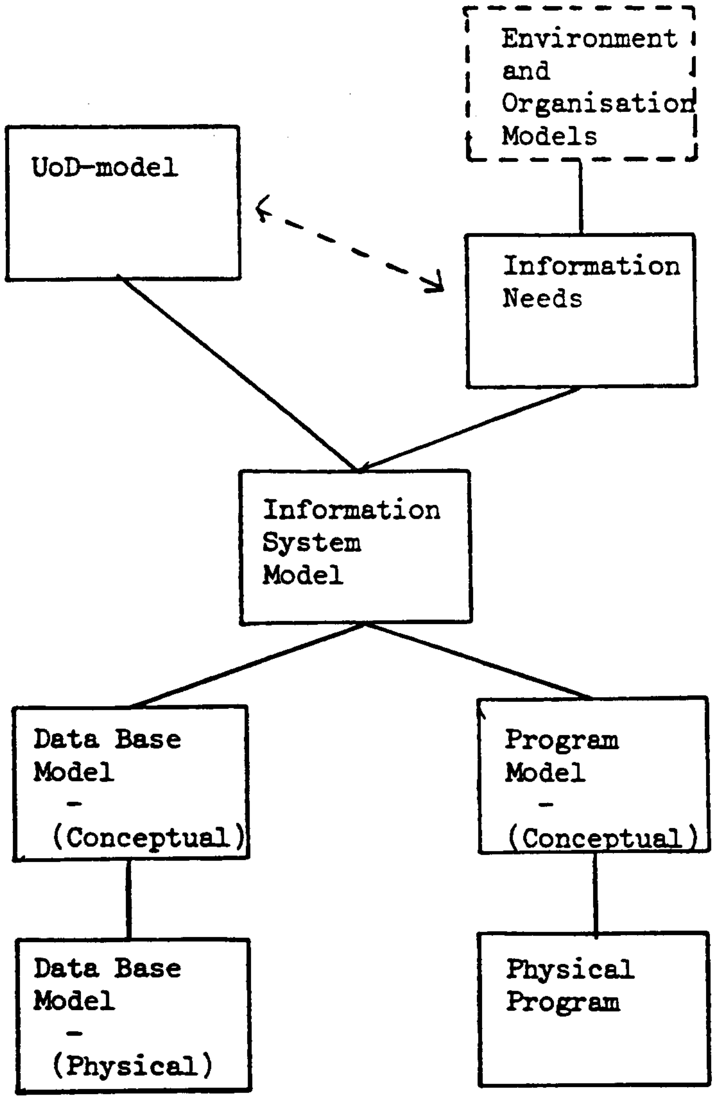

fig 3.1

To make the distinctions above more clear I will give an example. At Statistics Sweden we carry out regular surveys of the fishing industri in Sweden. The semantical aspects can be modelled by an object-oriented approach, where typical objects are fishermen, boats and equipment. Such a model is discussed in Malmborg-83. A pragmatically oriented model for an IS handling this survey could e.g. model the flow of information from data acquisition, via questionnaire, data entry, data editing, tabulation to publication. This type of model concerns activities at Statistics Sweden, which are certainly not performed by fishermen or boats. Based on such a model the information needs of the involved persons might be modelled. Note that neither model is a model of the planned computerized IS, but will later be used to create such a model. A deeper pragmatical analysis would rather concern the use of the produced statistical information inside and outside Statistics Sweden.

In a recent book (Langefors-84), Professor Börje Langefors, who has been very influential on the Infological Approach and the Statistics Sweden system development methods writes:

"In order to support decisions or other activities of human beeings, data must be consistent with the language and frames of references of the people that are to be served by the data. In other words: data must inform the people in the intended way.

Here it is important to remember that this information or comprehension problem has two aspects.

One of these is the linguistical (semantical) comprehension. The people must know the definition of the concepts referred to by the data. This is rather obvious, but has not been paid much attention to in many systems. Authorities, computer experts, accountants, and others tend to disregard this problem.

The other aspect is at least equally important, but may be even more difficult to take into proper account. It concerns comprehension in a deeper sence than the semantical one. Not only the phenomenon that the data (and the associated concept) gives information about, is important, but also the importance that this phenomenon has for those who are to use the data, for example, the consequences or possibilities that are implied by the reported phenomenon, and the actions that the information may lead to.

The first aspect (the linguistical-semantical) must of course be satisfied, if the latter, deeper aspect of comprehension is going to work - the human, pragmatical, or hermeneutical comprehension." (end citation)

Everyone involved in methods for large scale system development seems to argue for better communication between system analysts and different user groups (subject matter specialists and end users). However, there is no agreement on what they should communicate about. My answer to this last question is:

- They should start by building together models of aspects outside the IS

- There are two separate aspects to be modelled - semantical and pragmatical.
- It is not important which aspect to start with. The first models are anyhow crude.
- When the model of the IS (the functional specification) is to be built both models will be used, and they need to be consistent with each other. One typical way to reach consistency is to express the information needs of different users in terms of the semantic models.

In the current debate on conceptual (or semantic) models there seem to be two schools based on different aims, and presenting models of two different categories.

The first school aims mainly at a model to be used as a communications tool between analysts and users (subject-matter specialists and end users). The models are normally not formal or complete enough to be used as a reference model by the final running application. This is typical of simple object oriented approaches (The "SCB Systemeering Model") (SCB-79), SASMO (SAS-80), Object-Event Analysis (Olenfeldt-77). This type of approaches have been in practical use in Sweden for many years, and their popularity seems to be growing.

The second school finds a higher level of formalization absolutely necessary, and is more closely related to the aims of a "conceptual schema" (e.g. ISO-82). Most logic-based approaches belong to this category (CIAM, e.g. Bubenko-81, Gustavsson-82) and Interpretative Predicate Logic (ISO-82) are examples.

My approach tries to combine the ideas of the two schools into one coherent methodology.

Some conceptual model approaches choose not to make a distinction between a model of the UoD (outside the IS) and a model of the IS in conceptual (or semantic) terms. The same model can thus be seen to satisfy both aspects. Another interpretation could be that there are two models with 1-1 mapping between them. In OPREM (the proposed approach) the distinction is important, since the mapping normally is not 1-1. The mapping is controlled by the different information needs, as is illustated in Malmborg-84 (c.f. the summary of Malmborg-84).

Recent literature on conceptual modelling (e.g. Mylopoulos-84) often stress the common aim of modelling and AI, Data Models and High-Level Programming Language design. I think this is important. Many programming languages have a strong emphasis on modelling, where the possibility of executing the models without further transformations can be seen as an additional bonus. Examples are APL, Prolog, Simula and Smalltalk.

One framework for structuring methods for system development is based on ISO-82 and is discussed in Malmborg-83. The discussions on semantical versus pragmatical aspects above is based on this earlier approach by the author (cf. the summary of Malmborg-83).

A second alternative framework is the infological framework with 4 "problem areas". This framework has been used with somewhat different terminology by Langefors, Sundgren, Kahn and others. The first two areas are called "infological" or "user-oriented". The last two areas are called "datalogical" or "computer-oriented"

My framework is a refinement of the infological framework, where the semantical/pragmatical aspects have been separated. This separation is made to emphasize that the aspects are not to be ordered, but are complementary. In a corresponding manner the datalogical area has been split in two.

An alternative framework presented in e.g. Persson-86 is:

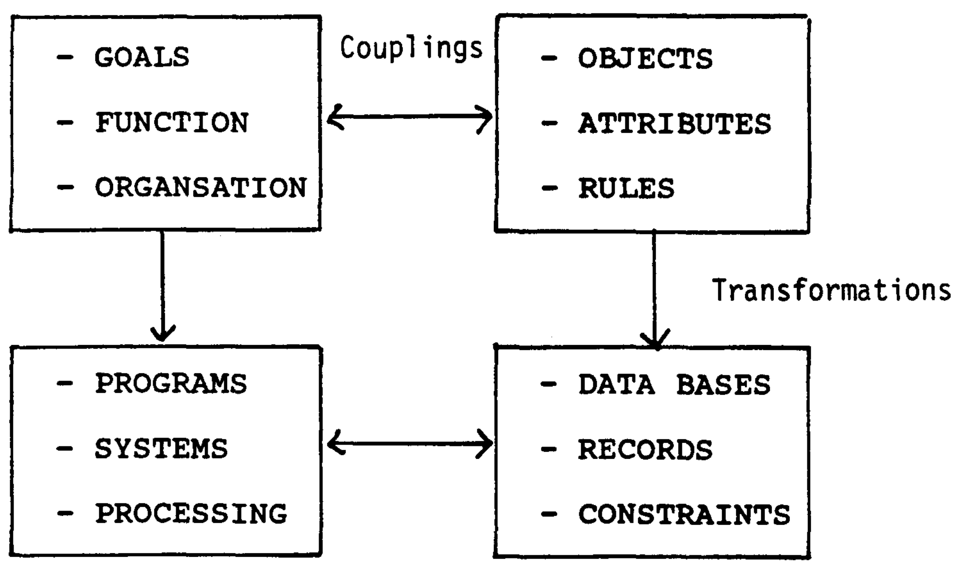

fig 3.2

This framework has also two problem-oriented and two computer-oriented areas. Note that my framework is rather close to this framework, but includes a common Information System Model. I consider the "4-window"-model above to be over-simplified. In Malmborg-83 (cf. the summary) the same semantic model is used to derive the structure of both programs and data bases.

3.2 The further development of the SCB model

The existing SCB model is not described in this paper. Descriptions can be found in Malmborg-82, -83 and -84 (cf. the summaries). Sundgren-84 (part C,D) is the recommended tutorial. The existing SCB model is based on what today is called conceptual modelling. The particular modelling technique, which is being taught and practised, is essentially situation or state oriented. A conceptual, or "infological" model, as described graphically, gives a static view of the objects and their relationships. Although there are some steps in the methodology, which explicitly consider such entities as births and deaths, events, and transactions, there is no complete analysis, let alone graphical representation, of the dynamical aspects of the object system (universe of discourse).

For many systems that a statistical office has to design, a static view is by and large adequate and sufficient. A statistical survey often aims at a "snapshot" of some phenomenon. In these cases the present SCB approach is fully adequate, and is supported by relevent software. In many other cases, e.g. when production of statistics is based on administrative, event-based registers, the SCB approach still works as well as known alternatives, but the models can not be used in the same exact way as a base for specifying all of the information needs, and the required processing.

In the present version of the SCB model there is a certain element of transaction analysis. During this analysis one analyzes among other things which transactions are going to change the contents of the data base of the planned system. In OPREM this analysis has been generalized to cover more completely the dynamics of the objects, properties, and relations in much the same way as this is done in the methodology developed by Michael Jackson (JSD, "Jackson System Development", a development of JSP, "Jackson Structured Programming", Jackson -83, -75). JSP/JSD is not described in this paper, but some material can be found in

Malmborg-82, -83, -84. The first part (60 pages) of Jackson-83 is the recommended tutorial. There is one important distinction between OPREM and JSD. OPREM starts from a traditional object system model (as used in the SCB approach above), and uses the more sophisticated analysis only when this is required. JSD, on the other hand, explicitly defines the situation or state oriented applications (a major class of applications at SCB, as I mentioned above), as being "outside the area of interest for the method". The traditional SCB-approach and JSD can thus be seen as complementary, and it should be fruitful to combine them.

There are two different ways of modelling events in the present version of the SCB model. If the users of the planned information system are interested in storing and processing information about a certain event, it should be regarded (also) as an object. Otherwize it should (only) be treated as a transaction, which affects the status of other objects, but is not in itself an object. In some cases this distinction is somewhat dubious, since it may be a question of implementation whether information about an event should be stored or not.

In OPREM the users/designers are encouraged to use a rather broad object concept when starting the modelling work. In a second phase a more structured model is then created by narrowing down the original, say, 5-25 objects (or rather object classes or object types) to, say, 1-7 objects. This can be done with a gain (rather than loss) of semantics by giving the latter objects an internal structure, which the original ones were not explicitely modelled to have. The internal structure can be modelled, like in JSD, as a tree structure.

I will illustrate the discussion above with a couple of examples.

Example 1. Let us assume that the planned information system should handle salaries and fringe benefits for the employees of an organization. One of the "fringes" is that managers get a leasing car. This means that a lot of special information has to be handled for the particular category of employees who are managers. The first attempt by users/designers to model this situation may be the following one:

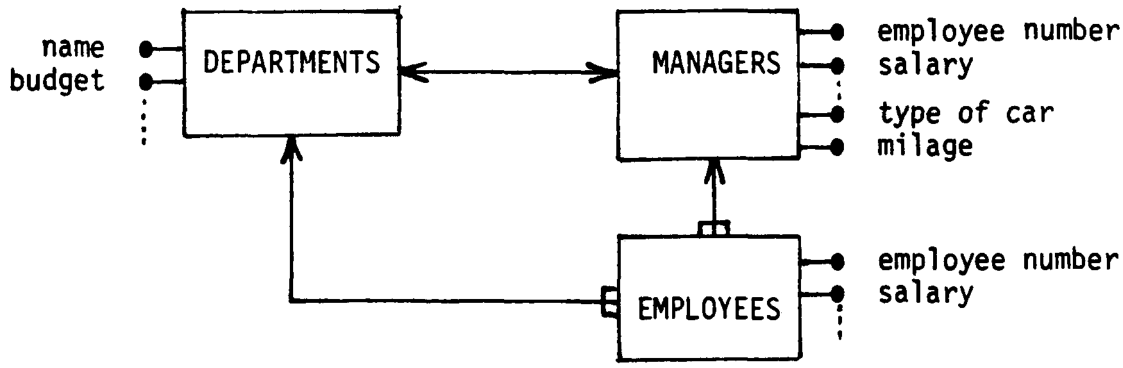

fig 3.3

One of the check-questions to be used during the second phase of the object system modelling is whether there are any objects (in the first version of the model) that have some properties in common. If so, these objects should be brought together into a "generic" hierarchy. OPREM uses the selection notation of JSP/JSD for such hierarchies. Thus in the example we get:

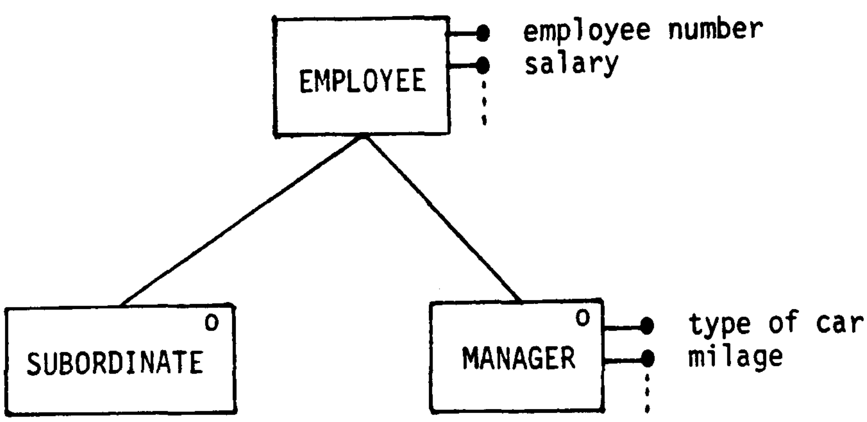

fig 3.4

The diagram should be read: Every object of the type EMPLOYEE is either a SUBORDINATE or a MANAGER. All EMPLOYEES have the properties "employee number" and "salary". Only MANAGERS have the properties "type of car" and "milage". On the highest level in the hierarchy we have the most general object type. All objects on a lower level inherit the properties of higher level objects. Thus in this example a manager will be understood to have the properties "employee number" and "salary", although this has not been explicitly represented in the diagram.

After the second phase of the analysis , the object system model will consist of (a) an overview graph, and (b) detailed subgraphs. Hierarchies like the one above are shown as detailed subgraphs. The overview graph will only contain top level objects from generic hierarchies. Thus in the example the overview graph will look like this:

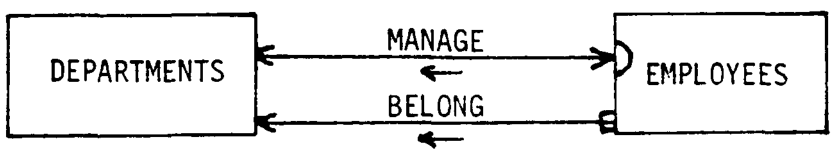

fig 3.5

The notation represents among other things the fact that only a subset of EMPLOYEES participate in the MANAGE relation. Such a relation is called a partial relation.

We can also note a difference in semantics between the two last diagrams above. The overview graph uses nouns in the plural form, indicating sets of objects, whereas singular forms are used in the generic hierarchy, indicating object types. One may also say that the overviews are on an extensional level, whereas the hierarchical graphs indicate intensional structures. In conceptual schemas the corresponding distinction exists between "iss-relations" (is subset) and "isa-relations"(is a).

Example 2. Let us assume that the users in our previous application require that the system should be able to handle and save all changes in salaries in order to enable a correct calculation of the holiday salary to be made once a year. We must also consider that an employee may have several employments during the year with different employment and salary conditions. In this case the first version of the object system model may be:

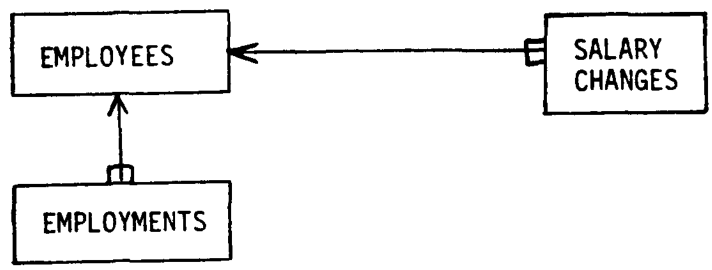

fig 3.6

In a more detailed analysis of the SALARY CHANGES (which have a dual object/event role) and their relations to (and effects on) the other object, a "career" model, or life history diagram, or entity structure may be useful:

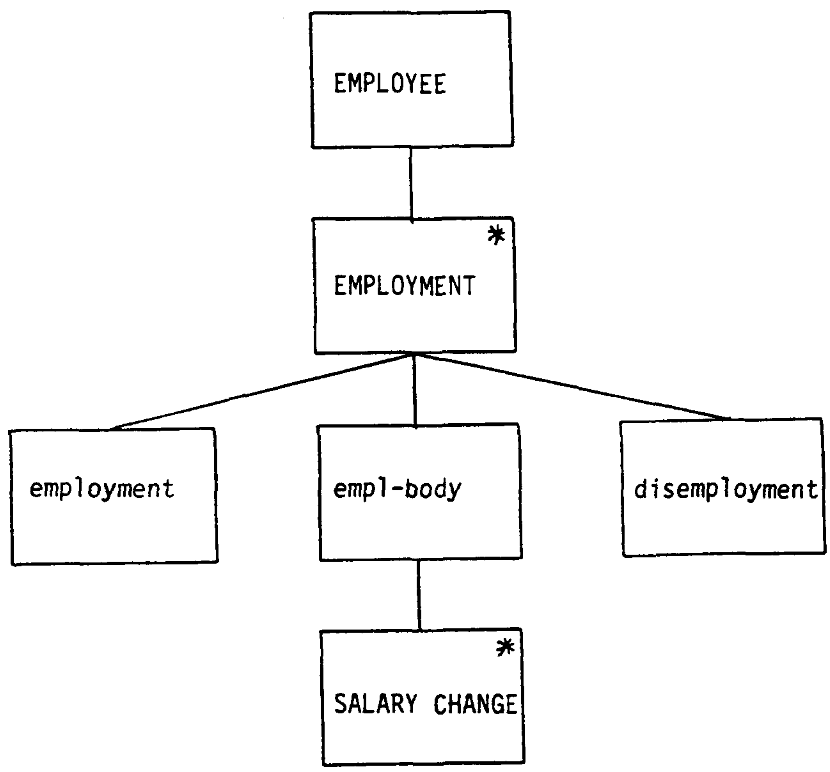

fig 3.7

This diagram, which has also borrowed its symbols from JSP/JSD, should be read as: Every EMPLOYEE goes through a career consisting of a number of EMPLOYMENTS . During each EMPLOYMENT there are a number of SALARY CHANGES. Each EMPLOYMENT starts with an employment event and ends with a disemployment event.

In a more detailed model of the dynamical aspects of the object system, we can see how e.g. the property "salary" is affected by the different events in combination with the history of the object. It is important that the "pragmatical" aspects are added after the basic semantic model is made. The property "salary" of EMPLOYEE is "pragmatic" because it is needed to express some information needs (queries, reports ..).

In certain more complex cases there are events in the object system which affect several different objects. In these cases a special structure is created, where the (event) object is subdivided into cooperating (internal) subevents, each of which affects only one object. This is discussed in Malmborg-82, section 2.5.

The two examples in this section have concentrated on two different aspects of a refined analysis of the first (overview) model of the object system. Sometimes there is a need for a combination of these two refinements, and a general life history structure ("entity structure" in JSD) will contain both selections (o) and iterations (*). By means of systematical transformations (Jackson-85) this kind of model can be turned into procedural programs, which can be building blocks in the final application system. In other cases the structural diagrams will be more useful as a basis for the data base design (Malmborg-84).

A problem with OPREM-P, as described above, is the lack of suitable tools. Manual implementation, using the transformations of JSP/JSD is tedious and the resulting code is thought to be ugly and "unstructured" by many programmers (cf. the discussion on program inversion in Malmborg-82, section 2.3). The different attempts on software tools are described in section 3.4. This ends the presentation of OPREM-P.

OPREM-D has been the working name for a version of OPREM where the JSP/JSD transformations (e.g. program inversion) are avoided. From the beginning the aim was to find systematic (possibly computerized) ways to transform from OPREM-P to OPREM-D, but this has not been succesful. Some of the ideas in this direction are described in section 3.3. The present approach to OPREM-D is rather unformalized, but is thought to gain some qualities by avoiding the use of computer tools! To give an idea of OPREM-D, I will continue with the example above. The notation used will not be JSD (as in Jackson-83), but the notation introduced in Malmborg-84.

If we convert fig 3.7 into pseudocode we get:

```
BEGIN
  DO WHILE employments
    employment;
    DO WHILE salary changes
      salary change;
    ENDDO
    disemployment;
  ENDDO
END
```

This is only a translation of the figure. In JSD and in OPREM the model is thought to be connected to the real world by a data stream. The real world EMPLOYEE-0 writes a data stream C which is input to the model process EMPLOYEE-1 (the notation is from JSD):

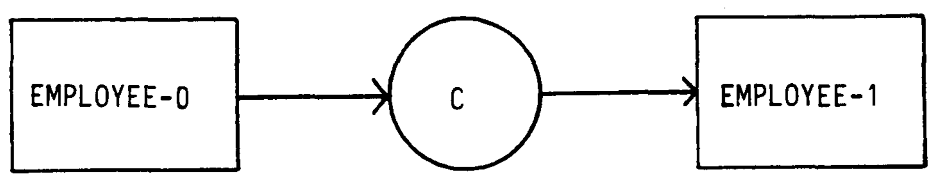

fig 3.8

Since there is a message in C for each action of (or affecting) the real world employee, the model process can simulate the real-world process exactly and syncronously. The events involved are:

- Employment. It is assumed that a salary is decided on employment.

- Salary change

- Disemployment

The structure text of the EMPLOYEE-1 process would be:

```
BEGIN
  open message stream;
  read message;
  DO WHILE employment (or "until dead" !)
    handle employment;
    read message;
    DO WHILE salary change
      handle salary change;
      read message;
    ENDDO
    handle disemployment;
    read message;
  ENDDO
END
```

Note that this basic model is free from error handling. It handles the prescribed ordering of events. The model doesn't yet include any attributes. These are added during the "pragmatic" analysis of information needs (or "functions" in JSD terminology). Malmborg-84 (pp 11-13) contains an example of this. The model above is a pure semantic model based on an analysis of events in the object system (or "UoD"). In JSD the model would be "inverted" and the corresponding program called from a transaction handler when events are reported to the system. Each read statement would correspond to a value of the state variable QS.

In OPREM-D we use the model for identifying the states of the objects. Each read statement corresponds to a state. The initial read can often be discarded as it corresponds to the state "not yet created". These states are given names, and their relevance is discussed with the "user" (subject matter specialist). In our example the states are:

- employed, but no salary change

- employed, and salary changed at least once

- disemployed

If the first two states should be kept distinct is a semantical question to be discussed at design time. If all employments begin with a trial period before they become permanent we might prefer 3 distinct states, otherwise the two first might be combined into one. The latter case is assumed.

After this analysis the state transitions are analyzed:

state / event		employed	disemployed
employment	:	ERROR	employed
salary change	:	employed	ERROR
disemployment	:	disemployed	ERROR

The different attributes and corresponding operations (e.g. change salary) can be added to this "concequence matrix". This matrix is then a base for generating code. The basic consequence matrix with objects and events is part of the existing SCB approach. It is used to describe the effects of the events on the objects. The adding of states and state transitions to the matrix is a proposed extension of OPREM-D.

Observe that state oriented development is nothing new, and e.g. Martin-86 anf Mills-87 indicate a new wave of interest. The key issue in OPREM-D is the approach to decide on the state-space on a pure sematical base before the (pragmatical) information needs are analyzed.

3.3 Remarks on the transformation from an OPREM-P type model to an OPREM-D type model

I will give a broad picture of the development of "structured" concepts for program and system developement. This will hopefully help in understanding the difference between OPREM-P and OPREM-D.

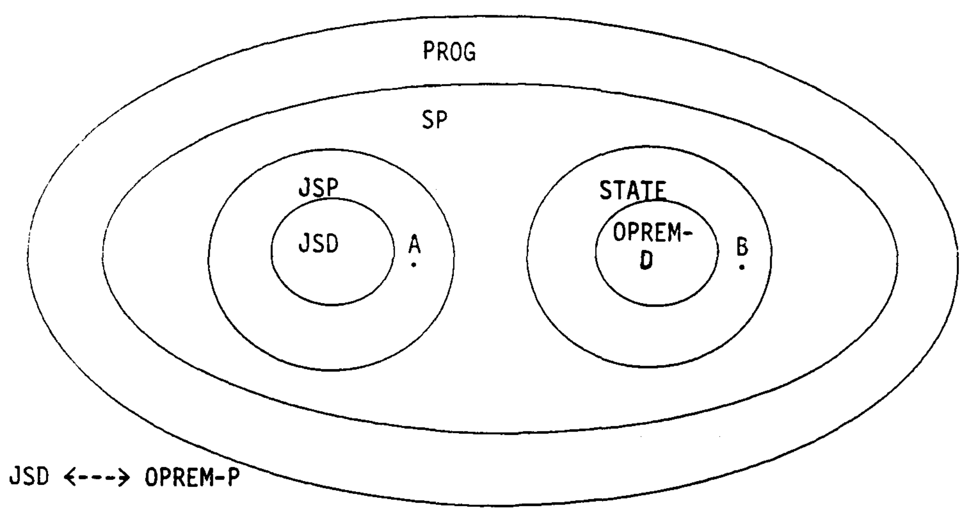

fig 3.9

The figure above is a Venn-diagram (set-model) of possible programming solutions for a given problem (or set of problems). The different areas represent:

- PROG is the set of all programs. I include "systems" with several programs connected by files. This is a broad space as most maintenance programmers are aware of.

- SP is the subset of "structured" programs in the classical meaning. They are built from a restricted set of control structures (sequence, selection, iteration). Cf Boehm-66, Mills-72, Dahl-72.

- JSP is the small subset of SP corresponding to systems resulting from application of JSP. The selections and iterations are based on the "semantics" of the problem, excluding most of the SP-solutions. Typically the JSP solutions have a nested iteration structure.

- JSD is the subset of JSP-type programs resulting from applying JSD with its stricter semantic orientation. Note that JSD is not suitable for all programming problems. Only certain, but important, classes of systems are suitable for the application of JSD. For these classes OPREM-P and JSD generate the same set of solutions.

- STATE is the set of state-oriented programs. They typically have a main loop handling state transitions (from a state table). This set is disjoint from the JSP set in all but trivial cases. Typically the programs have a flat loop structure.

- OPREM-D is the subset of state-oriented programs with a semantic base of the same type as JSD.

As an example take the problem of parsing a context-free language. The JSP-solution is identical to a recursive descent parser (A in fig 3.9) with its close coupling between the productions of the language definition and the control structures of the parser. A parser-generator of type YACC generates a state-oriented program (B in fig 3.9) solving the same problem (Aho-86, or other texts on compiling).

Considering the transformations performed by YACC it would be reasonable to find a transformation OPREM-P to OPREM-D. However, YACC type transformations generate large state-spaces not suitable for OPREM-D. I have not found this approach fruitful.

Another idea is to perform the JSP/JSD program inversion (cf. 2.3 in Malmborg-82 and Jackson -75, -83), and then apply some technique similar to partial evaluation (Haraldsson-77) to derive the processes for the different values of QS. These processes can then be used in a state-based program. In practical situations it is rather clear how the result would look and, the proposed mechanism seems to be overcomplex.

The result of analyzing different approaches was to abandon the transformation approach and go for the direct approach described above in section 3.2.

3.4 On the needs for CASE-tools supporting OPREM-P and OPREM-D

Most of the work described in Malmborg -82, -83, -84 deals with OPREM-P. The need for computer-based tools for OPREM-P was obvious at an early stage. Much of Malmborg -82, -83, -84 can be seen as specification work for (a possibly ADA-based) tool. However, the resources for developing such a tool were never available, but some experimental tools were made:

- A simple system was developed for "Infological simulation" where the user could enter events and study the result (cf. the dialogue in section 4.5 of Malmborg-82). The system was developed on an Apple II personal computer. The tool was used to demonstrate some of the ideas for the planned tool.

- A program generator ("SEL-83") was developed in UCSD-Pascal (Malmborg-84b). This was a test of methods for creating program generators. Generating programs (in e.g. the C programming language) was, and is, a key element of planned tools.

- A system for demonstrating key concepts on cooperating processes in OPREM-P was partially implemented. This tool should demonstrate that a suitably designed environment (or operating system) would eliminate the need for transformations in JSD and OPREM-P. The system was based on the PASCAL-S compiler by N Wirth (Berry-81), extended to handle several cooperating programs . This was technically made by a redesign of the s-code interpreting system. I have later found a similar approach in Ben-Ari-82 where it is used to demonstrate "principles of concurrent programming".

Note that these tools were made for demonstration and training in the "tool-makers trade". I have not had the programming resources for making a full-scale development.

The system specified in Malmborg-88 is partly based on these earlier ideas, but has another purpose. However, it is a long-range goal to include the OPREM-D concepts in the system and thus reach an OPREM-D tool.

4. An introduction to Statistical and Scientific Data Base Management (SSDBM)

The two papers Malmborg-86 and Malmborg-88 are submissions to the working conferences on Statistical and Scientific DataBase Management (III SSDBM in Luxembourg and IV SSDBM in Rome). As these papers (and conferences) are somewhat specialized some basic knowledge on SSDBM are assumed from the reader of the papers. In the context of this thesis I do not want to make such assumptions. Hence I will now present some introductory material.

The most important and most widely known framework in the area of SSDBM is presented in the paper "Statistical Databases: Characteristics, Problems and some Solutions" by Arie Shoshani of Lawrence Berkeley Laboratory (Shoshani-82). Arie Shoshani is one of the founders of the SSDBM conference series. Shoshani-82 is a broad overview covering:

- Some example databases

- Characteristics of SSDBM

- Physical organization

- Optimization

- Logical Modelling

- User Interfaces

- Integrating statistical analysis and data management

- Security

I will here present some material on characteristics of SSDBM, Physical organization and logical modelling.

One characteristic of SSDBM is that data belong to two types:

- Category attributes contain categories for measured data in a scientific database.

- Summary attributes contain the measured data. Statistical summaries (and analysis) applies to these attributes

Note that:

- Category attributes may serve as a composite key for the summary attributes

- In many statistical data bases all possible combinations of the category attributes (i.e. the full cross product) exist. In such cases each value of a category attribute repeats as many times as the product of the cardinality of the remaining category attributes (c.f. fig. 4 in Malmborg-86).

- The range of category attributes is usually small, from as little as two (e.g. "sex") to a few hundreds (e.g. oil type). Often category attributes are grouped together so as to have fewer categories, such as using "age groups" rather than "age". Often, coded versions of the text are assigned to long category values.

In a presentation oriented statistical data base the same pattern exists. Category attributes correspond to classification criteria (or "gamma"-variables). Summary attributes correspond to the indicators to be retrieved (or "beta"-variables).

The characteristics above in combination with the types of information needs (statistical measures, statistical tables) imply that traditional DBMS become technically unsuitable for statistical data bases. In the extreme case a commercial DBMS is optimized for for the efficient retrieval of one record, i.e. the values of all attributes for one key-value. A typical retrieval situation for SSDBMS is to calculate a measure (mean value, deviation ..) or make a statistical table based on one or a few attribute(s), for all or a large subset of the objects (population). Typical access methods for SSDBMS include:

- transposed files (i.e. "attribute partition" or "vertical partition")

- compression of category atttributes by run-length encoding (or more advanced techniques with the same aim, cf. Svensson-79, Karasalo-83)

- matrix storage and access by "array linearization". this implies that the values of the category attributes are not stored at all in the database, instead they are represented as meta-data.

Logical models are often based on the criteria and storage philosophies above. Malmborg-86 contains introductory material on logical models for statistical data bases, and I refer the reader to this paper.

Enc. 1 Summaries of the reports constituting the thesis

Summary of "The OPREM-approach, An extension of an OPR-approach to include dynamics and classification". (S/SYS-E12 or Malmborg-82).

This paper was originally prepared as a submission to the CRIS-1 (Comparative Review of Information Systems) conference (Olle-82). However, it was not accepted. Like all other submissions to this conference the paper is based on the "CRIS-case". This case is still beeing used for comparing different approaches to system development. CRIS-solutions can be found in Olle-82, Wigander-84, Olle-86, and in several other papers. I think most method developers have used the CRIS-case to test their ideas. Comparative analysis of CRIS solutions can be found in Malmborg-83, Olle-83, Maddison-83, Olle-86, Verrijn-Stuart-87.

Section 1 on "General background and rationale" is mainly superceded by this overview. It should be noted that even the "broad" part of this thesis makes some important delimitations. This is discussed at the end of section 1, where it is stated that:

- The goal is to analyze and develop the possibilities of this postulated approach (an extended OPR-approach). This is a clear positioning on the scale of "semantic richness" (Nilsson-79).

- A distinction is postulated between "data" and "meta-data". The data will in no way automatically change the model (meta-data). This is a delimitation to traditional system development in comparison with some classes of intelligent knowledge bases.

Further delimitations are presented in the summary of Malmborg-84.

Section 2.1 is a short decription of the SCB-model of 1981. It is shown how the information needs (statistical tables) can be expressed using the semantic model by using "$\underline{\alpha\beta\gamma}$-analysis". Sections 2.2 and 2.3 decribe JSP/JSD as of 1981 (JSD was only an embryo in Jackson-75). Section 2.4 is to motivate the need for integration. Sections 2.5 - 2.7 describe the OPREM-P approach of 1981. This is to be seen as a description of the model to reach. The method for reaching this model is presented in Malmborg-84. Section 2.8 tries to show that the proposed set of extensions is adequate.

Section 2.9 is a summary of the proposed tool and of the method.

Section 3 consists of discussions on some other relevant topics. 3.1 is a first introduction to the OPREM-P use of ADA as a modelling language. The rich type-concept in ADA is used for specification of static aspects. The specifications are grouped into objects by the package-concept. The task-concept is used for dynamic aspects. Note that this and the other ADA-examples were made during the work of specifying a (possibly ADA-based) tool for system development using OPREM-P.

Section 3.2 is a discussion on early versus late binding. This is central to object oriented development and object oriented programming languages where some form of late (or "interpreted") binding of procedure calls is needed.

Section 3.3 stresses the point that there need not be any simple correspondance between what is modelled as data (or information) and what is modelled as procedures (processes) on one hand and what is technically realized in a running system as data and procedures respectively on the other hand. This is a basis for considering OPREM-P and OPREM-D as dialects of the same approach.

Section 4 is the IFIP-case solution. The use of the semantic model for "JSP-type" specification of information needs is presented in 4.6.

The specification for the IFIP-case can be found in an appendix.

Summary of "An analysis of systems design methodologies using the ISO framework" (S/SYS-E14 or Malmborg-83)

This paper was published in the proceedings from the "Second Scandinavian research seminar on information modelling and data base management" (Tampere 1983). The distinction between "Universe of Discourse" (UoD), "environment" and "information system" from ISO-82 is the base dimension used for comparing different methods. Relating to section 3 of this overview the correspondance between:

universe of discourse < -- > semantical aspects
environment < -- > pragmatical aspects
information system < -- > data base and programs

should be noted. It is the framework from Malmborg-83 that has evolved into a more general framework.

Another central dimension in the analysis is the temporal complexity of the modelling situation (based on the information needs). Section 2.2 of Malmborg-83 is a treatment with examples.

The two dimensions above are used to analyze various methods (CRIS-submissions and some Scandinavian approaches).

Section 4 is on "two approaches to handle the static/dynamic dichotomy". This is an attempt to contrast the dynamic approach of JSD and OPREM-P with the declarative approach of Bubenko et. al (Bubenko-81, Gustafsson-82)

The declarative approach is seen to handle the static/dynamic dichotomy in the UoD by focussing on "static". The "dynamic" aspects of the UoD are not seen as such, but are handled as e.g. existence specifications and other "constraints".

In dynamic approaches (JSD, OPREM-P) the dynamics of the perceived UoD is recognized and dynamic conceptual models are used. The dynamical changes of attributes are modelled explicitly. The "static" aspects are seen to be represented by the "state vectors" of the dynamic model.

Summary of "Stepwise formalization of information systems specifications by extending a simple object-oriented approach" (P/ADB - E20 or Malmborg-84).

This paper was originally presented at the Seventh Scandinavian Research Seminar on Systemeering in Helsinki 1984, and was published in the proceedings.

The theme of the conference was "User Participation: Strategies, Methods and Tools", and the paper has one section each on strategi, methods and tools. In the introduction and problem delimitation some further delimitations are presented (c.f. the summary of Malmborg-82):

- The application to be developed is too complex to suit end user tools of type simple data base handlers ("Card-Box type systems) or spreadsheets (type

Visi-Calc or Lotus 1-2-3)

- It is assumed that there is a specific organisation with users, who need the application. Specifically all "Systems programming" is excluded.

The strategy is for the "user" and the "systems developer" to communicate while specifying and designing the system. What they should communicate about is discussed in the section on strategies. Section 3.1 in this overview is based on the same ideas. The fig 3.1 in section 3.1 is systematically derived from different "pattern of models" in Malmborg-84. Generally the first part of Malmborg-84 is recommended as an introduction to my thinking on system development.

The section on Methods in Malmborg-84 partly duplicates earlier discussions on modelling situations, the existing SCB approach and JSP/JSD. A significant addition however is the presentation of how to transform an infological model into an "OPREM" model. Malmborg-84 is the first presentation of the approach in section 3.2 of this overview (but oriented towards OPREM-P rather than OPREM-D)

The second part of the section on Methods can be seen as further work on specifying (the possibly ADA-based) tool for OPREM-P. The issues described are on handling "complex dynamics" and illustrate among other things that the mapping of fig (3.2) (data model -> database, process/event-model -> programs) is too simplified. In the example the event/transactions based model is transformed into a data base model (schema)

In the final section on tools there are some further discussion on the planned OPREM-P tool.

Summary of "On the semantics of aggregated data" (U/ADB-E25 or Malmborg-86)

This paper was originally presented at the "Third international workshop on Statistical and Scientific DataBase management" in Luxembourg July 1986, III SSDBM.

The paper contains "An overview of conceptual modelling". The simple taxonomy presented consists of 4 dimensions:

- What is modelled (the reality or a database)?

- Should the model be set-oriented or type-oriented?

- What basic concepts should be used (ER, EAR or predicate logic)

- Shall we have "predefined" relations with a specified semantical meaning (e.g. isa-relations)

These dimensions are used as a frame of reference for the rest of the report.

The section on "The development at Statistics Sweden" gives a background to the development at Statistics Sweden. The influences from the early work on the "Infological Approach" (Sundgren-73,75) on the AXIS Statistical DataBase Management System (SDBMS) are analyzed. The SCB Systems Development Approach is also based on this early work.

The section on "comparison with other research and development" compares the SCB approach for Statistical databases (as realized in the AXIS system) with some other systems:

- SUBJECT (as in Chan/Shoshani -82)

- GRASS (as in Rafanelli/Ricci -83)

- STRAND (as in Johnson -81)

- LASD (as in Lutz -84)

- CANTOR (as in Karasalo/Svensson -83)

The final section is "A new proposal for the semantics of aggregated data". This is the first presentation of the ideas behind the system described in the next paper (Malmborg-88)

Summary of "Design of the user-interface for an Object-oriented Statistical database. (R&D Report 1988:11 or Malmborg-88)

This paper was originally presented at the "Fourth international working conference on Statistical and Scientific Database Management" in Rome June 1988 (IV SSDBM).

The first section on "The meaning and goal of an Object-Oriented Statistical Database" (OOSD) is a discussion on different traditions leading to a possible concept of an OOSD:

- Traditional databases handle records(segment types). If they are extended to handle more complex objects (drawings, pictures, matrices, ...) they become "object oriented".

- If we have object oriented languages (like Simula-67, Smalltalk ..) and make the objects "persistent" by storing them on secondary storage we get object-oriented databases.

It is suggested that a true object-oriented database should fulfill criteria from both traditions.

The second section is on "the development towards Macintosh-style user interfaces", and is a history of that development.

The third section is on "the interaction between statistical query languages and meta data". The proposed form of interaction is to have a meta-database that is "browsed" in search of descriptions for the needed data. The meta data thus found, can directly be used for specifying information needs towards the data base. An approach of this type is used by the Japanese Land Agency as described in Sato-86. Some aspects of this Japanese approach is presented.

The sections 4-6 describe the proposed software system consisting of:

- A graphical meta data browser. (sec. 4)

- A table design language, TBE-2 (sec. 5)

- A model for the interaction between the parts above (sec.6)

REFERENCES

Aho, V.A., Sethi, R., Ullman, J.D. (-86)
Compilers Principles, Techniques, and Tools
Addison-Wesley

Ben-Ari, M. (-82)
Principles of concurrent programming
Prentice/hall

Berry R.E. (-81)
Programming Language Translation
Ellis Horwood Limited, Chichester

Birtwistle, G.M., Dahl, O-J., Myrhaug, B. & Nygaard,K.
SIMULA BEGIN
Studentlitteratur, Lund, 1973 or
Petrocelli/Charter, New York, 1975

Boehm (or Böhm), C., Jacopini, G. (-66)
Flow Diagrams, Turing Machines and languages with only two formation rules.
Communications of the ACM, May 1966

Brodie, M.L. (-84)
On the development of Data Models in Brodie-84b

Brodie, M.L., Mylopoulos, J., Schmidt, J.W. (-84b)
On Conceptual Modelling - Perspectives from Artificial Intelligence, Databases, and Programming Languages.
Springer-Verlag, 1984

Bubenko jr, J.A. (-79)
On the role of 'understanding models' in conceptual schema design
Proc 5th Int Conf on Very Large Data Bases, Rio de Janeiro 1979

Bubenko jr, J.A. (-81)
On Concepts and Strategies for Requirements and Information Analysis.
SYSLAB Report No 4

Bubenko jr, J.A. (-86)
Information System Methodologies - A Research View
SYSLAB report No. 40, also in CRIS-3

Burstall, R.
(see Darlington, J.)

BYTE (-81)
Theme on Smalltalk
BYTE, Aug 1981

Cameron, J. (-83)
JSP/JSD Tutorial
IEEE Computer Society Press (Catalog No EHO206-3)

Chan, P., Shoshani, A. (-80)
SUBJECT: A directory driven system for organizing and accessing large statistical databases.
Proc 6th Int Conf on Very Large Data Bases 1980

Chen, P.P-S. (-76)
The Entity-Relationship Model - Toward a unified view of Data.
ACM Transcactions on Data Base Systems, March 1976

Codd, E.F. (-79)
Extending the Database Relational Model to Capture More Meaning
ACM, Transactions on Database Systems Dec, 1979

CRIS-1,2,3 see Olle -82, -83, -86

Dahl, O.J., Dijkstra, E.W., Hoare, C.A.R. (-72)
Structured Programming
Academic Press 1972

Darlington, J. (-85)
Program Transformation
(Article in BYTE, Aug-85 - refers to earlier work by Burstall and Darlington)

Dijkstra, E.W.
Notes on Structured Programming
in Dahl-72

Furtado, A.L., Kerschberg, L. (-77)
An Algebra of Quotient Relations
Proc ACM SIGMOD Int Conf on Management of Data, 1977

Goldkuhl, G., Nilsson, A., Röstlinger, A. (-80)
Att specificera informationssystem på ett användarorienterat och systematiskt sätt.
(Försöksupplaga 1979, förlag okänt!)

Goldkuhl, G. (-81)
Information Modelling Considered as Language Formalization
SYSLAB, Stockholm University

Graves, R.B.
Data Base Modelling as an Aid to Data Editing
Statistics Canada 1979, also proc ISIS-79, Bratislava

Gustafsson, M.A., Karlsson, T., Bubenko jr, J.A. (-82)
A declarative approach to conceptual information modeling
SYSLAB Report No 8, also in Olle-82

Haraldsson, A. (-77)
A program manipulation system based on partial evaluation
Linköping Studies i Science and Technology Dissertations No 14

Hewitt, C.
Viewing Control Structures as Patterns of Passing Messages
MIT AI Memo No 8, 1977

Hoare, C.A.R. (-78)
Communicating Sequential Processes
Communications of the ACM, Aug-78

Hugosson, M-Å. et al (-83)
MBI-metoden. En metod för verksamhetsanalys
Studentlitteratur 1983

ISO-82
Concepts and terminology for the conceptual schema and the information base
ISO TC97/SC5/WG3 1982

Jackson, M.A. (-75)
Principles of program design
Academic Press, London 1975

Jackson, M.A.
Structural Design of the Information System
Michael Jackson Systems Limites, 1977

Jackson, M.A. (-83)
System Development
Prentice Hall, 1983

Jansson, C.G. (-86)
Taxonomic Representation
SYSLAB, University of Stockholm, Aug 1986

Johnson, R.R. (-81)
Modelling Summary Data
Proc ACM SIGMOD Int Conf on Management of Data, 1981

Kahn, L., Grubbström, A.
Elementarmetoden
Department of Information Processing, Stockholm University

Kay, A.O.
Microelectronics and the Personal Computer
Scientific American Sept 1977

Kerschberg, L., Klug, A., Tsichritzis, D. (-76)
A taxonomy of data models
University of Toronto

Lagerlöf, B., Malmborg, E. (-85)
Användarorienterad Systemutveckling vid SCB; Dagens erfarenheter och framtida inriktning
RDF-konferens Vägen till bättre informationssystem 85-10-23--24
also as SCB, SYS-info 85:19

Langefors, B. (-73)
Theoretical Analysis of Information Systems
Studentlitteratur/Auerbach, Lund

Langefors, B., Sundgren, B. (-75)
Information Systems Architecture
Petrocelli/Charter, new York 1975

Langefors, B. (-84)
Systemering med applikationsgenerator exemplifierad med DataFlex
Studentlitteratur 1984

Lindencrona, E.
A study on conceptual data modelling
University of Gothenburg, 1979

Lundeberg, M., Goldkuhl, G., Nilsson, A. (-79)
Systemering
Studentlitteratur, Lund

Lutz, H. (-84)
Experiences in data structuring gained from running a general statistical data bank system.
Statistical Journal of the United Nations, ECE 2 (1984) 179-190.

Maddison, R.N., Baker, G.J. Bhabuta, L., Fitzgerald, G., Hindle, K., Song, J.H.T., Stokes, N., Wood, J.R.G. (-83)
Information system methodologies
Wiley Heyden Ltd on behalf of the British Computer Society

Malmborg, E. (-82)
The OPREM-Approach - An extension of an OPR-approach to include dynamics and classification.
Statistiska Centralbyrån S/SYS-E12

Malmborg, E. (-83)
An analysis of systems design methodologies using the ISO-framework.
Second Scandinavian research seminar in information modelling and data base management (Acta Universitatis Tamperensis ser B vol 19, Tampere 1983), also as SCB S/SYS-E14.

Malmborg, E. (-84)
Stepwise formalization of information systems specifications by extending a simple object-oriented approach.
Seventh Scandinavian research Seminar on Systemeering (helsinki-84), also as SCB P/ADB-E20

Malmborg, E., Chowdhury, S. (-84b)
Programgenerering som metod för produktion av portabla standardprogram - ett experiment och en metodansats.
SCB, ADB-METOD 1984:04

Malmborg, E. (-86)
On the Semantics of Aggregated Data
Proc Third Int Workshop on Statistical and Scientific Database Management, Luxembourg 22nd-24th July 1986, also as SCB, U/ADB-E25

Malmborg, E. (-88)
Design of the User-interface for an Object-oriented Statistical Database.
Statistical and Scientific Database Management, Fourth International Working Conference SSDBM, Rome, Italy, June 1988, Proceedings, Lecture Notes in Computer Science 339, Springer-Verlag 1989.

Martin, J.
Information Engineering: A Management white paper.
KnowledgeWare Inc, 1986

Mills, H.D. (-72)
Mathematical foundations for Structured Programming.
IBM, 1972

Mills, H.D.
Box structured information systems
IBM Systems Journal, No 4 1987

Myloupolos-84
An overview of Knowledge representation
in Brodie-84b

Nelson, T. (-80)
Articles on Object-oriented languages
Creative Computing, Oct-80

Nilsson, B. (-79)
On Models and Mappings in a Data Base Environment
SCB, URVAL nr 9

Nilsson, G. (-84)
Table by Example - TBE
SCB, Feb 1984

Olenfeldt, L. (-77)
Objekt/Händelseanalys
STU-Report 77-4936 (Swedish board for technical development)

Olle, T.W. et al (-82)
Information Systems Design Methodologies: A Comparative Review
North-Holland 1982

Olle, T.W. et al (-83)
Information Systems Design Methodologies: A Feature analysis.
North-Holland 1983

Olle, T.W. et al (-86)
Information systems Design Methodologies: Improving the Practice
North-Holland 1986

Peckham, J., Maryanski, F.
Semantic data models
ACM Computing Surveys, sept 1988

Persson, A.
in Statskontoret-86

Rafanelli, M., Ricci F.L. (-83)
Proposal of a Logical Model for Statistical Data Base
Proc 2nd Int Workshop on Stat Database Man

Riksdataförbundet-80
Informationsmodellering
RDF Rapportserie 10:8

Robinson, K.A.
An entity/event data modelling method
The Computer Journal, Aug 1979

SAS-80
SASMO - SAS Data Models
SAS Data planning group, STOXS, Aug 80

Sato, H., Nakano, T., Fukazawa, Y., Hotaka, R. (-86)
Conceptual Schema for a Wide-Scope Statistical Database and its Applications.
Proc 3rd Int Workshop on Stat Database Man, Luxembourg, 1986

SCB-79
SCBs systemeringsmodell
SCB, S/SYS, kompendium 79-03-02

Shipman, D.W.
The functional data model and the data language DAPLEX.
ACM Transactions on Database Systems, March 1981

Shoshani, A. (-82)
Statistical Databases: Characteristics, problems, and some Solutions.
Proc 8th Int Conf on Very Large Data Bases, 1982

Smith, J.M., Smith, D.C.P
Data Base Abstractions - Aggregation and Generalization
ACM Transactions on Database Systems, June 1977

Statskontoret-81
Att välja ovh införa metoder vid systemutveckling, - tillvägagångssätt, -praktikfall
Statskontoret rapportserie 1981:17