Conceptual Description for Information Modelling Based on Intensional Containment Relation

Hannu Kangassalo

Department of Computer Science, University of Tampere,
P.O.Box 607, FIN 33101 Tampere, Finland
Internet: hk@cs.uta.fi

Abstract. In information systems design the concepts by which the Universe of Discourse (UoD) is described, must be recognised, named, and described before the system can be constructed. The description of concepts depends on points of view of users, on their knowledge about the UoD, and on their goals. Concepts are constructed from knowledge primitives and from other concepts by using an intensional containment relation as a basic relationship. In this work we analyse some properties of the intensional containment relation, proposed by R. Kauppi, study some properties of concept description language Concept D based on that relation, and conclude by describing briefly the formal structure of a conceptual schema made by using Concept D. We also give some empirical evidence about the structure of a conceptual schema of this kind.

1 INTRODUCTION

In the design of information systems more and more responsibility of the work is given to users. Problems are expressed by domain-oriented concepts and rules. The conceptual structure of the system as well as the content of data bases must be defined by using concepts which the user uses in his work. The ultimate edge of this evolution is that users can work with the information system as if working with concepts of the Universe of Discourse (UoD) alone, without being confused by implementation oriented concepts of the system. A conceptual description of the subject matter, including the corresponding information system, is called a conceptual schema.

The conclusion to be made of this evolution from the user's perspective is that we should replace the whole information system with the conceptual schema of the UoD, supported with facilities for:

- accessing, analysing, changing, manipulating and maintaining concepts (conceptual descriptions) from which the conceptual schema describing the UoD has been constructed , and
- accessing, manipulating and maintaining data corresponding to the conceptual schema.

This paradigm is based on the view that a conceptual schema defines a systematic 'theory' of the UoD. The concepts are constructed on the basis of goals and business rules of the user organisation. The notion of a theory is to be understood here on the basis of the non-statement view according to which a theory is a mathematical structure together with its intended applications [8]. That is, the theory is not a set of statements but an abstract construct composed of knowledge primitives, concepts, and associations between them. That view makes it easier to relate the conceptual schema closely to the structure of the system of concepts applied in a certain field, without linguistic biases. It can also deepen our knowledge of the process of conceptual modelling in general and especially modelling as the theory construction view in particular [9].

In the following a concept is defined to be an independently identifiable structured construct composed of knowledge primitives and/or other concepts. Concepts are not classified e.g. into entities, attributes, or relationships. This kind of classification is not an intrinsic feature of knowledge - it is a superimposed abstract scheme into which knowledge is often forced. There are various schemes which are commonly used. The application of several different schemes gives rise to the problem of semantic relativity. We try to avoid it.

A concept is connected to a set of rules of application, which specify the conditions in which the concept can be meaningfully applied. Concepts in a theory are connected with associations. The rules of application of concepts in the theory, together with associations connecting the concepts, form the set of rules of application of the theory.

The concepts, the structure they form and the behaviour of the system should be described from the point of users, not of the computer system. All implementation-oriented constructs must be hidden from the user. The user should be able to analyse the content of concepts completely, if necessary, from the point of view of his role in the domain.

On the other hand the user should be able to modify the conceptual schema when the changes of the UoD make it necessary, or when he wants to change his own conception of the UoD.

In this work we analyse first the structure of information requirements and the problem of satisfying them. Then we analyse briefly construction of concept and a specific relation of intensional containment by which concepts can be constructed. Then we present some features of a concept description language Concept D and discuss about properties of a conceptual schema made by using Concept D.

2 INFORMATION REQUIREMENTS

A person, who needs information, has first to recognise the situation in which he wants to specify his information requirements, then to select of what kind of information he wants, and to formulate the requirements on the basis of the properties of the situation and his needs. In this phase he has to evaluate the relevance of the information he requires. After that, he has to solve the problem of how the information can be produced and finally to produce the information, and use it in order to reach his goal. The situation is described in Figure 1.

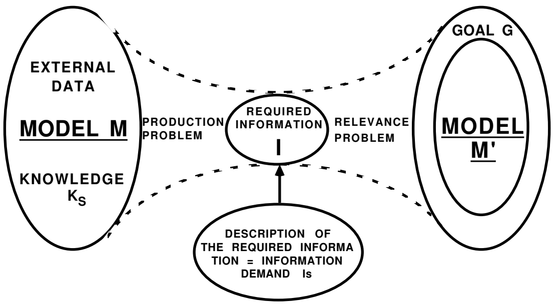

Figure 1. The situation for satisfying information requirements

At the beginning of the process of satisfying the information requirements of the user S the situation can be described as follows. Model M and model M' are conceptual schemata or theories applied in the situation. They can be represented by using several different structuring principles and expressed by using some concept description language, e.g. CONCEPT D [2, 3].

Model M describes the current part of the UoD. It consists of knowledge K_S of the user(s), and the external data available for them which are relevant for the situation of satisfying information requirements. Descriptions of the conceptual content of external data should be available, either in the minds of users, in data repositories, or in the conceptual schemata. Otherwise users cannot understand the meaning of data. The recognition of the current part of the world requires, or may require, knowledge acquisition and modelling. Modelling involves concept construction and analysis.

Model M is, in its primitive form, a simplified description of the UoD, constructed by using ordinary or simplified concepts of person S, and from his point of view. A more advanced model may contain concepts which belong to some theoretical structure describing the UoD from certain point of view. The construction of model M may require concept construction, analysis, and synthesis, too.

Model M' describes the future situation, which should appear according to the goal G specification of person S. The recognition of the future situations usually requires knowledge acquisition, concept formulation, conceptual modelling, and possibly analysis of epistemic changes, too. The goal specifies the conditions under which model M' should occur. It contains things like: 1) time, when the situation in M' should occur, 2) costs for reaching M', 3) the plan for reaching M'.

The required information I is what is wanted by the user in order to reach his goal. It contains e.g. the data (values) with the desired meaning, descriptions of new concepts, (partial) conceptual schemata, and possibly analysis of consequences of proposed epistemic changes. Information demand I_S is the description of the required information on which the user really works.

If the information demands can be stated, there is still the question of how the data carrying the required information can be produced. A solution of the production problem indicates how the information can be produced from external data and users' knowledge K_S. A relevance problem is a problem of how to recognise that the required information helps in reaching the situation described in model M' in G, is the required information relevant for reaching the situation in model M', and how can we find the information which is relevant for reaching the situation in model M'.

The models M and M' are not necessarily constructed by using the same ontological principles, i.e. they do not belong to the same structuring principle class. If they belong to different classes, it may be necessary to search for transformations which make it possible to change concepts, conceptual constructs, and conceptual schemata and data from one structuring principle to another. It also may require epistemic changes to be accomplished.

When the information requirement of the user has been satisfied, model M can be expanded by adding the original information demand into it, and the required and produced information as well. They can be used in solving the future information requirements.

3 STRUCTURE OF CONCEPTS

Concepts are created by a human mind. Understanding and use of information requires that concepts and conceptual constructs by which the UoD is described are properly understood. The user must know and understand the knowledge structure of concepts, not only names of words used to refer to this structure. An approach in achieving that knowledge and understanding is that the user analyses the content of concepts. He should go from the description of a concept to the description of another, lower level concepts until the level of observable concepts is reached. In order to analyse, develop, and communicate concepts with other people we have to externalise them by using a concept description language.

Concepts and rules defining the UoD are made visible and shareable in a conceptual schema. At the same time the conceptual schema defines the meaning and semantic structure of data to be stored into the data base.

A graphic conceptual schema supported by functionalities for data manipulation can be seen as the information system itself. It is a representation of a concept space in which a user can navigate and perform operations on concepts. He can also store, retrieve, and manipulate data based on the conceptual schema. A good support system is necessary. The user interface must concentrate on constructs of the conceptual level.

The graphical conceptual schema language, Concept D, is a visual language that supports the development and definition of a conceptual schema and the use of the database [2, 3]. The language is based on the intensional approach to conceptual modelling. The system of concepts is based on the use of an intensional containment relation, the interpretation of which is extended from the traditional interpretation. Concepts are defined as structured units. A prototype system was implemented at the University of Tampere in a project at 1984 - 1988. It was demonstrated at the ER-conference in Rome 1988. A new version is being developed now.

The language contains three sub-languages which are closely related: one for knowledge acquisition about concepts used in the UoD, one for describing conceptual schemata, and one for interacting with conceptual schema and the database. A visible conceptual schema supports users in recalling and understanding the conceptual structure of the UoD. It also facilitates the user to analyse and study concepts of different levels. It hides the database from users so that they work only with concepts of the UoD. It is used as a basis for putting queries to the database, too.

Knowledge is composed of knowledge primitives. Knowledge primitives are in their basic form the smallest structural units of knowledge. There are several types of knowledge primitives, each having a corresponding cognitive capability in human mental processes. The following are some of the most common knowledge primitives:

- Name of the concept used to refer to a concept.
- Intensional relationship between two concepts.
- Extensional relationship between occurrences of concepts.
- Identifying property is a property of concept B intensionally contained in concept A that enables an occurrence of concept B to be used to identify an occurrence of concept A.
- Condition schema.
- Constraint schema.
- Value set is a set of other concepts and their representations associated with a given concept to represent it.
- Function is mapping from a value set to another.
- Semantic rule is a text explaining the concept .

For each type of knowledge primitive there is a corresponding graphical or textual representation which makes knowledge visible. Knowledge primitives can be combined according to certain rules to make up larger constructs of knowledge - concepts.

The fundamental notion, *concept*, is regarded as central epistemological unit of human knowledge. The knowledge content of a concept is its *intension*. The concepts, knowledge primitives, and the structure they form in the intension of a concept are called its *characteristics*. A set of objects (as well as data representing objects in the data base) to which a concept applies is called its *extension*. A concept has always the intension, but its extension can be empty. The elements of the extension are called *occurrences* of that concept.

We need an additional term to refer both to knowledge primitives and concepts. A *knowledge unit* is either a knowledge primitive or a concept.

Concepts are always based on the *point of view of one or several people,* but almost never of all people. A person recognises and uses characteristics which are important for him from his own point of view, and constructs concepts on the basis of them [1, 2, 3]. That fact results in a need of integration of concepts which belong to different points of view of different people in the data base design.

A *basic concept* is a concept which cannot be analysed using other concepts of the same conceptual system. It can contain one or more knowledge primitives.

The working hypothesis used in this work is that the basic epistemological relation between knowledge units is the relation of *intensional containment* [5]. The relation of intensional containment is a binary relation defined on the set of knowledge units. The first member of each tuple in the relation must be a concept. The second member can be a concept or a knowledge primitive. Note, that concepts and knowledge primitives are abstract entities which contain knowledge about occurrences.

The relation of intensional containment (IC) holds between concept A and knowledge unit P, if and only if knowledge unit P is one of the characteristics of concept A. Then we say that concept A contains intensionally knowledge unit P. It means that the knowledge forming concept A intensionally contains the knowledge forming knowledge unit P. Knowledge unit P is a part of the knowledge forming concept A.

For example, a concept of doctor is known to most people. They also know that a doctor is a person. The knowledge forming concept of doctor contains intensionally the knowledge forming knowledge unit of person. However, the concept of doctor contains intensionally also the knowledge that this person has a special education needed to be a doctor.

That concept A intensionally contains knowledge unit P is symbolised as follows [5]:

$$"A \geq P"$$

The inverse relation of IC between knowledge units P and A is 'being intensionally contained'.

The relation of intensional containment is reflexive, transitive, and antisymmetric [5]. Therefore it is a *partial ordering* on the set of knowledge units. Partial ordering expressed as intensional containment relationship can be regarded as a basic relationship in organising knowledge units.

IC shows the fundamental organisation of knowledge in concepts and in a conceptual schema on the basis of knowledge content of concepts. That it is not the same thing than how the definitions of concepts have been organised or ordered in the schema. We are talking about the knowledge required to recognise phenomena in the UoD, not how the definitions of these concepts are constructed. A definition may contain knowledge which is not intensionally contained to the defined concept. That fact emphasises that a concept and its definition are separate, different things.

Let us suppose, that concept A contains intensionally concept B, i.e. $A \geq B$. Concept B is often in some sense more general than concept A, i.e. concept A contains more and different information than concept B.

The relation of intensional containment is often expressed in words by using a phase "is-a" and it is also held that is-a is the only interpretation of that relation. An example could be: a doctor is-a person. However, "is-a" is a specialised type of intensional containment relation. It is also often used quite ambiquously. There are other relations between knowledge units, that meet the properties of partial ordering, too. It must be noticed that these relations may have more meaning than a simple partial ordering.

It must be observed, too, that a physical object, or an extensional object, and concepts used to characterise it are different things. In human mind a physical object is represented as observational data, which is based on information received from sense organs. All concepts are abstracted constructs made by using abstraction functions either from observational data or other (simple or complex) constructs of knowledge.

In a natural language there are several types of expressions that formally meet the properties of an intensional containment. Some examples of these expressions and their conceptual content are described and analysed in the following:

1. '- is-a -', e.g. a concept of driver of a car contains intensionally a concept of person, i.e. a driver of a car is-a person. In this case, the relation has two additional properties: it *preserves* the object described by the concept and *changes the*

epistemological abstracness of the conceptual content of a concept describing the object. A series of concepts shows an example: man ≥ person ≥ living_thing ≥ entity. It may be that in two concepts describing the same object there is not a single common characteristic recognised. That fact may cause difficulties in integration of conceptual schemata.

2. 'Contains', or 'has a component', e.g. a concept of car contains intensionally a concept of engine, or a concept of car has a component of concept of engine. Two different additional properties are: it *limits* the object described by the concept and *preserves the level of epistemological abstraction of the conceptual content of concepts describing the object*, i.e. the extensional object is diminishing, but the epistemological abstraction level of concepts used to describe the object remains the same all the time. An example: car ≥ engine ≥ ignition system ≥ distributor. Abstraction level is rather low in this case.

The relations above are clearly based on the relation of intensional containment. In a natural language there are expressions which behave like the relation of intensional containment, except in the use of them transitivity often reduces to triviality because in occurrences of these relations there are usually only two levels.

3. 'is' or 'has', e.g. a car is red, or a person has a name, i.e. concept of car contains concept of colour, or concept of person contains concept of identifier (or name). The words 'is' or 'has' indicate that a concept has a relation of property to another concept, e.g. car ≥ specific colour ≥ colourful.

We can conclude, that the relation of intensional containment includes the other relations as special cases.

A *definition* of the concept is a rule or a linguistic instruction which specifies how the knowledge forming the defined concept (definiendum) is to be constructed from the knowledge given in the defining concepts and in the definition itself. In order to find out the intension of the defined concept a definition must in some cases be evaluated, i.e. the definiendum must be actually constructed. A *derived concept* is a concept the characteristics of which have been derived from the characteristics of other concepts in the way described in the definition of that concept [3].

The defining concepts are in turn defined by other concepts until the level of undefined, basic concepts is reached. A basic concept cannot have other concepts as its characteristics. Structurally the derived concept is always a directed acyclic graph based on the relation of intensional containment. The graph contains the name of the definiendum and all information needed to define it. In other words, the graph contains the definition of the definiendum.

Intensionally contained concepts are defining concepts, but all the defining concepts shall not necessarily be contained concepts. In addition, the graph contains IC relations which relate the name of the definiendum to defining concepts, and possibly to knowledge primitives which add some characteristics to the defined concept, e.g. an identifying property may not be derivable from defining concept but it introduces new knowledge into the defined concept.

There is an infinite number of different definition types. The type of the definition specifies how the characteristics of the definiendum are actually to be derived from the characteristics of defining concepts. In practice only few definition types are used. The most commonly used types are:

1. *Aggregation*, in which a concept is defined as a collection of its characteristics.
2. *Generalisation*, in which a concept is defined as a collection of those characteristics that all its defining concepts have in common.
3. *Value transformation*, in which a concept is 'defined' by specifying how the values representing it can be derived from values representing the defining concepts. Observe, that the intension of the definiendum is not constructed in the definition; it must be evaluated separately.

A definition and the defined concept are different things. Several different definitions may all evaluate to the same concept. This is important from the methodological point of view, and also from the users' point of view, e.g. in the integration.

A *concept structure* is a diagram which represents a definition of a concept. A concept structure consists of a defined concept and of its definition hierarchy, in which the properties of the definiendum derive from the properties of basic concepts [2]. The graphic layout is meaningful in a concept structure diagram. The definiendum is on top of the hierarchy and concepts defining it are on the next or lower levels of the hierarchy. Concept structure diagrams are used to define concepts of the working environment of users.

Accordingly, the conceptual schema of UoD is a concept structure (definition) of the single concept defining the whole UoD. It contains intensionally definitions of all concepts recognised in the UoD, as well as all additional knowledge primitives. For example, in an insurance company the conceptual schema of a new life insurance system is a concept structure which has the definiendum 'Life insurance system' and which contains all conceptual information needed to specify the whole system.

In aggregation a definiendum is constructed by composing two or more characteristics together and by assigning a name to the resulting construct. At least one of the characteristics must be a concept. Usually there are several defining concepts in the definition. They are considered to be connected by a logical connective AND. In the definition all the known relations between contained concepts must be specified, as well as all inscriptions attached to them. The general graphic pattern of aggregation is in Figure 2. For simplicity the symbols for exclusive-or and for identifiers have been omitted. Symbol ° indicates that there can be any number of this characteristic (from 0 to n) [2, 3].

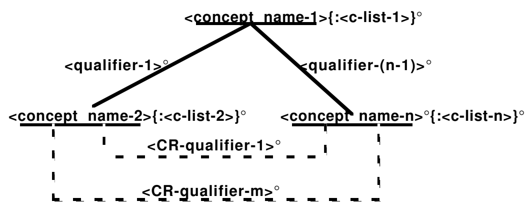

Figure 2. Graphic pattern of aggregations.

The defined concept is considered to be on a higher level than defining concepts. A qualifier can be a condition list, a constraint list or a conditional constraint. A CR-qualifier can be a constraint list or a conditional constraint. They may concern e.g. values of occurrences, value sets, extensions of concepts, equality or differences between occurrences and occurrence time or occurrence conditions of concepts [2, 3].

An occurrence of the concept exists if an occurrence of every defining concept exists and all constraints in the definition are true. For a particular occurrence of the definiendum, the occurrence of a defining concept can be missing if the condition attached to the corresponding intensional containment line is false.

In generalisation a definiendum is constructed by assigning to it either all the characteristics that the defining concepts have in common, or some subset of these characteristics, see Figure 3.

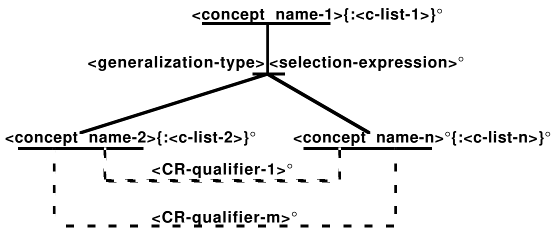

Figure 3. Graphic pattern of generalisations.

Definition by generalisation differs from the definition by aggregation in one essential aspect. The definition must be evaluated before the intension of the definiendum is revealed. It means simply that an implicit concept is made explicit. The resulting concept is structurally just an ordinary concept, e.i., the aggregate.

An example of the use of definition by generalisation is in Figure 4 [2] and the result of evaluation is in Figure 5.

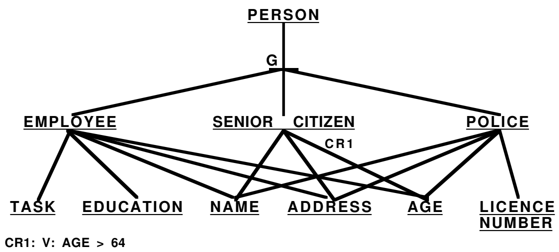

Figure 4. Definition of concept PERSON.

Figure 5 shows that all information concerning the concept will be under the name of the concept. In complicated schemata that feature is very useful.

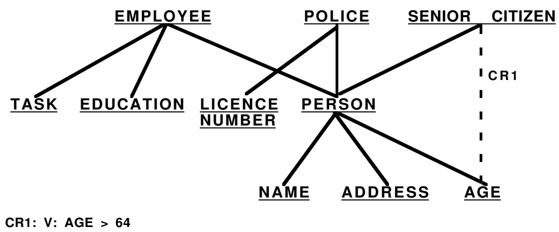

Figure 5. Result of evaluation of the definition of concept PERSON.

In value transformation a definiendum is constructed by specifying how the value representing the definiendum is derived from the values representing the defining concepts. The transformation is defined as a function included into the 'definition'. The general graphic pattern of value transformation is in Figure 6 [2, 3].

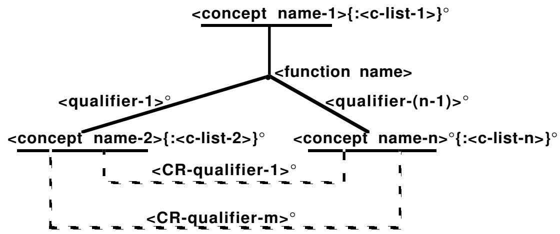

Figure 6. Graphic pattern of value transformations.

Conditions and constraints are used to add structural rules to the concept definition. A condition is used to indicate structural alternatives in the intension of the concept. A constraint is used to select the set of possible occurrences of the concept. They may appear as a qualifier, CR-qualifier, or as an attribute in a c-list.

The language for expressing conditions and constraints is simple. A statement consists of a type specification and a condition or a constraint specification. The condition specification and the constraint specification consists usually of a rather simple Boolean statement.

The type indicates how the statement should be interpreted, i.e. which type of characteristic it concerns. An example is in the Figure 7 [2, 3].

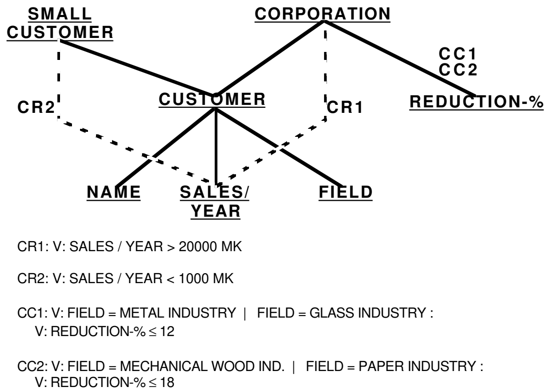

Figure 7. An example of the use of constraints and conditional constraints.

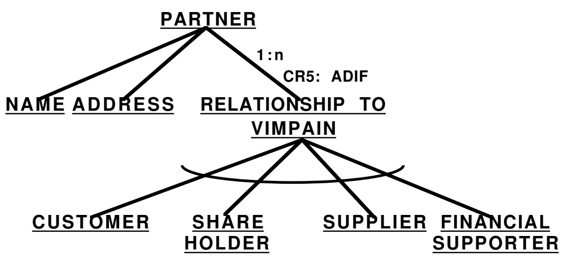

Figure 8. Relationships between Vimpain and its partners.

In Figure 8 a combination of an identity constraint and an exclusive-or constraint is presented. Vimpain is a company which has partners. A partner may have several relationships to Vimpain (1:n), which must be different. That is indicated by an identity constraint CR5: ADIF (all different).

Figure 8 indicates also how alternative concepts can be described. They are considered to be connected by a logical connective OR.

Defining concepts with the identifying property are called *identifiers*. An identifier can be simple or compound. An identifier is effective in the whole definition hierarchy of the definiendum, i.e. its *scope* is the definition hierarchy of the definiendum. The scope can be limited by a scope limit symbol.

4 STRUCTURE OF A SCHEMA

An important topic in conceptual modelling is the structure of a conceptual schema (i.e. models M or M') applied in organisation of knowledge. In using original entity-relationship diagrams there was no additional structuring principles for organising the content of a conceptual schema used at all. As a consequence, big schemata were difficult to read and understand. Later, when abstraction type of generalisation was taken into use, the representation method, based on work of Quillian [7], was adopted. It shows general constructs at the top of the diagram and specialised constructs at the bottom of the diagram. However, that representation scheme seems to be inappropriate in many cases.

Very often interesting structures are in the middle or lower parts of the system. Some other ways of representation might support users better. The approach used in Concept D seems to be an interesting alternative. The structure of the schema, and the way how this structure is represented are different from the current ER-practice. It can be seen in Figure 9 of [3]. Theoretical justification for this difference has been received in [6, pp. 94-97].

Let C be a universe of concepts and let ≥ be the intensional containment relation between concepts belonging to C. A formal structure of <C;≥> is a meet semilattice, in which the general concept G is the bottom element of C. However, there is no such special concept in which every concept of C is intensionally contained, and therefore C has no top element. Also, there is no such concept which is a sup(a,-a), where -a is the intensional negation of the concept a. Thus, <C;≥> cannot be a complete lattice. Instead, it is a complete semilattice [6].

Empirical evidence in favour of the method used to organise and represent a conceptual schema in Concept D has been received in the project described in [4]. In an insurance company, from 144 concept structures describing information requirements of users a conceptual schema was developed by integrating them. As a result, the schema was received which contained about 1400 concepts, organised in the way described above. The schema contained 22 levels of hierarchy, from the highest level which contained only the name of the definiendum, 'Life insurance system', to the lowest level which contained only 3 basic concepts. A rule was followed that a concept is located on the level which is one level lower than in which it was used at the lowest. As a result, the characteristics of a concept are on a lower level than the concept itself, and it is used by concepts from the higher level than its own level.

Most basic concepts, about 400 of them, were on rather low levels of the schema, but quite many were also on top levels.

There were about 1000 derived concepts. Most of them were located on middle levels. For some concepts there were several specialisations which were located on a higher level than general concepts, e.g. a customer and an insurance, just as in Figures 5 and 7. Concept of a customer and an insurance, which are essential for the company, were located on quite low level, but not on the lowest level.

On higher levels there were quite many abstract but specialised concepts, e.g. concept of medical treatment, insurance policy, etc.

The manipulation of concepts requires that some concept operations are available, just like set-theoretical operations or relational operations. There are several proposals published. My current interest is in comparing these proposals. Concept operations should be usable for constructing new concept descriptions in such a way that detailed concept occurrences can be developed, as well as correct inferences can be made on the basis of them. There is a problem that practical concepts are usually rather complicated. In the current version of Concept D there are no real concept operations used. Instead SQL was used to implement them on a level hidden from users.

References:

[1]. Hautamäki,A. Points of View and Their Logical Analysis. *Acta Philosophica Fennica.* Vol. 41, Societas Philosophica Fennica, Ph.D dissertation, Helsinki 1986.

[2]. H. Kangassalo, CONCEPT D - A Graphical Formalism for Representing Concept Structures. In: H. Kangassalo, (Ed.), *Second Scandinavian Research Seminar on Information Modelling and Data Base Management.* Acta Universitatis Tamperensis. Ser.B, Vol.19, Tampere 1983.

[3] H. Kangassalo, COMIC - A System and Methodology for Conceptual Modelling and Information Construction. *Data & Knowledge Engineering.* 9 (1992/1993) 287-319, North-Holland.

[4]. H. Kangassalo and P. Aalto Experiences on User Participation in the Development of a Conceptual Schema by Using a Concept Structure Interface. In B.Shackel, (Ed) *Human-Computer Interaction - INTERACT'84.* North-Holland 1985.

[5]. R. Kauppi, Einführung in die Theorie der Begriffssysteme. *Acta Universitatis Tamperensis,* Ser.A, Vol.15, Tampereen Yliopisto, Tampere, 1967.

[6]. J. Palomäki *From Concepts to Concept Theory. Discoveries, Connections, and Results.* PhD Dissertation. Acta Universitatis Tamperensis. Ser. A, Vol. 416, University of Tampere, Tampere, 1994.

[7]. M.R. Quillan Semantic Memory. In Minsky,M.,(Ed) *Semantic Information Processing.* MIT Press, Cambridge, MA., 1968.

[8]. W. Stegmüller, *The Structure and Dynamics of Theories.* Springer-Verlag, Berlin, 1976.

[9]. Vadén,T. Conceptual Modelling as Theory Construction: Some Implications for the Metalevel Treatment. In Jaakkola, H., Kangassalo,H., Kitahashi, T., Márkus,A., (Eds.), *Information Modelling and Knowledge Bases V: Principles and formal techniques.* IOS Press, Amsterdam, 1994.