Methods of Conceptual Knowledge Processing

Rudolf Wille

Technische Universität Darmstadt,
Fachbereich Mathematik,
Schloßgartenstr. 7, D–64289 Darmstadt
wille@mathematik.tu-darmstadt.de

Abstract. The offered *methods of Conceptual Knowledge Processing* are procedures which are well-planed to mean and purpose and therewith lead to skills for solving practical tasks. The used means and skills have been mainly created as translations of mathematical means and skills of *Formal Concept Analysis*. Those transdisciplinary translations may be understood as transformations from mathematical thinking, dealing with potential realities, to logical thinking, dealing with actual realities. Each of the 38 presented methods is discussed in a general language of logical nature, while citations give links to the underlying mathematical background. Applications of the methods are demonstrated by concrete examples mostly taken from the literature to which explicit references are given.

Contents

1 Conceptual Knowledge Processing

Conceptual Knowledge Processing is considered to be an applied discipline dealing with ambitious knowledge which is constituted by conscious reflexion, dis-

R. Missaoui and J. Schmid (Eds.): ICFCA 2006, LNAI 3874, pp. 1–29, 2006.

cursive argumentation and human communication on the basis of cultural background, social conventions and personal experiences. Its main aim is to develop and maintain methods and instruments for processing information and knowledge which support rational thought, judgment and action of human beings and therewith promote the critical discourse (cf. [Wi94], [Wi97b], [Wi00b]).

The adjective *"Conceptual"* in the name "Conceptual Knowledge Processing" underlines the constitutive role of the thinking, arguing and communicating human being for knowledge and its processing. The term *"Processing"* refers to the process in which something is gained which may be knowledge or something approximating knowledge such as a forecast, an opinion, a casual reason etc. To process knowledge, formal elements of language and procedures must be activated. This pre-supposes formal representations of knowledge and, in turn, knowledge must be constituted from such representations by humans.

To understand this process, the basic relation between *form* and *content* must be clarified for Conceptual Knowledge Processing. A branch of philosophy which makes basic statements on this is *pragmatic philosophy* which was initiated by Ch. S. Peirce [Pe35] and is presently continued among others in the discourse philosophy of K.-O. Apel [Ap76] and J. Habermas [Ha81]. According to pragmatic philosophy, knowledge is formed in an unbounded process of human thinking, arguing and communicating; in this connection, reflection on the effects of thought is significant and real experiences stimulate re-thinking time and again. In this process, form and content are related so closely that they may not be separated without loss.

Theoretically, Conceptual Knowledge Processing is mainly founded upon a mathematization of traditional philosophical logic with its doctrines of concept, judgment, and conclusion. The core of the mathematical basis of Conceptual Knowledge Processing is *Formal Concept Analysis* [GW99a] which has been developed as a mathematical theory of concepts and concept hierarchies during the last 25 years. Although Conceptual Knowledge Processing deals with actual realities, it obtains its basic forms of thinking from mathematics that, according to Peirce ([Pe92]; p.121), has the aim to uncover a "great Cosmos of Forms, a world of potential being". Above all, Formal Concept Analysis as applied mathematics provides Conceptual Knowledge Processing with a rich amount of mathematical forms of thinking; this has been proven useful in a large number of applications. For such a success it is essential that conceptual representations of knowledge can be materialized so that they appropriately merge form and content of the processed knowledge.

As mathematical theory, Formal Concept Analysis with its notions and statements is strictly based on the common *set-theoretical semantics* which is grounded on abstract sets and their abstract elements. For the explanation of the mathematical notions, statements, and procedures in this paper, the reader is referred to the monograph *"Formal Concept Analysis: Mathematical Foundations"* [GW99a]. The notions and statements discussed in the framework of Conceptual Knowledge Processing shall be understood with respect to the semantics of their specific field of application. If they refer to different fields of application, their semantics has to be more abstract (eventually up to the philosophical semantics). To make the connections between Formal Concept Analysis and Conceptual Knowledge Processing clear, notions and statements of Formal Concept Analysis have to be transformed to suitable notions and statements of Conceptual Knowledge Processing and vice versa. For basic notions such transformation is given by the following list of correspondences (cf. [Wi05b], p.28f.)

Formal Concept Analysis	↔	**Conceptual Knowledge Processing**
formal context	↔	(logical) context
(formal) many-valued context	↔	many-valued context
(formal) object	↔	object
(formal) attribute	↔	attribute
many-valued attribute	↔	many-valued attribute
(formal) attribute value	↔	attribute value
formal concept	↔	concept
extent	↔	extension
object extent	↔	object extension
intent	↔	intension
attribute intent	↔	attribute intension
(formal) object concept	↔	object concept
(formal) attribute concept	↔	attribute concept
(formal) subconcept	↔	subconcept
(formal) superconcept	↔	superconcept
infimum of formal concepts	↔	largest common subconcept of concepts
supremum of formal concepts	↔	smallest common superconcept of concepts
concept lattice	↔	concept hierarchy

Based on such correspondences, this paper aims to show how Formal Concept Analysis gives rise to a spectrum of methods of Conceptual Knowledge Processing applicable for gaining knowledge for a broad variety of reasons and purposes.

2 Methods

In [Lo84], scientific methods are characterized in general as follows:

> A *method* is a procedure which is well-planned according to mean and purpose and therewith leads to skills for solving theoretical and practical tasks.

In the case of Conceptual Knowledge Processing, basic means and skills for its methods are mainly translations of mathematically defined means and skills of Formal Concept Analysis. Those translations interprete the mathematical means and skills with respect to actual realities so that they become understandable for common users in their specific semantics. In the sense of Peirce [Pe92], the *transdisciplinary translations* may be understood as transformations from mathematical thinking, dealing with potential realities, to logical thinking, dealing with actual realities (cf. [Wi01], [Wi05b]).

2.1 Conceptual Knowledge Representation

The mathematization of conceptual knowledge by Formal Concept Analysis is based on the understanding of *concepts* constituted by their extension and intension, respectively. For a concept, its *extension* contains all objects falling under the concept and its *intension* comprises all attributes (properties, meanings) common to all those objects. Thus, the representation of conceptual knowledge can be grounded on a *context* consisting of a collection of objects, a collection of attributes, and a relation indicating which object has which attribute. A context corresponds to a formal context which, in Formal Concept Analysis, is usually materialized by a cross table. Therefore, Formal Concept Analysis suggests the following elementary representation method of Conceptual Knowledge Processing:

M1.1 Representing a Context by a Cross Table: A context can be represented by a *cross table*, i.e., a rectangular table the rows of which are headed by the object names and the columns headed by the attribute names; a cross in row g and column m means that the object g has the attribute m. An example is given in [GW99a], p.18.

Conceptual knowledge is often represented by 0-1-tables. Then it is necessary to make explicit in which way the zeros and ones shall give rise to concepts. In the case that they lead exactly to the same concepts as the cross table in which the crosses are at the same places as the ones, such table is called a *one-valued context* (cf. M4.2) and considered as equivalent to the corresponding cross table.

M1.2 Clarifying a Context: *Object Clarification* of a context means to remove all objects except one in each class of objects having the same attributes. Dually, *Attribute Clarification* of a context means to remove all attributes except one in each class of attributes applying to the same objects. *Clarifying a Context* means to apply to a context both: Object Clarifying and Attribute Clarifying (cf. [GW99a], p.24).

A cross table of a context resulting from a clarification may be completed by inserting the name of each removed object g in front of the name of that object having the same attributes as g and inserting the name of each removed attribute m above the name of that attribute applying to the same objects as m. The completed cross table is often a considerably smaller and better readable representation of the original not clarified context than the cross table described in M1.1.

M1.3 Reducing a Context: *Object Reduction* of a finite logical context means first to apply Object Clarification to the context and then to remove each remaining object the object concept of which is the smallest common superconcept of proper subconcepts of that object concept. Dually, *Attribute Reducing* of a finite context means first to apply Attribute Clarification to the context and then to remove each remaining attribute the attribute concept of which is the largest common subconcept of proper superconcepts of that attribute concept. *Reducing a Context* means to apply to a context both: Object Reducing and Attribute Reducing (cf. [GW99a], p.24).

A finite context resulting from the reduction of a context is (up to isomorphism) the smallest context the concept hierarchy of which has the same hierarchical structure as the concept hierarchy of the original context. Thus, the finite reduced contexts are structurally the smallest (implicit) representations of finite concept hierarchies.

M1.4 Representation of a Concept Hierarchy by a Line Diagram: The concept hierarchy of a finite context can be visualized by a line diagram as follows: The concepts of the hierarchy are represented by small circles in such a way that upward leading line segments between those circles can indicate the subconcept-superconcept relation. Every circle representing a concept generated by an object/attribute has attached from below/above the name of that object/attribute (cf. M2.1). Those attachments of object and attribute names allow to read off the extension and intension of each concept from the representing line diagram: the extension/intension of a concept consists of all those objects/attributes the names of which are attached to a circle belonging to a downward/upward path of line segments starting from the circle of that concept (cf. [GW99a], p.23).

Unfortunately, up to now, no universal method is known for drawing well-readable line diagrams representing concept hierarchies. For smaller concept hierarchies, the method of *Drawing an Additive Line Diagram* (see [GW99a], p.75) often leads to well-structured line diagrams. This is the reason that quite a number of computer programs for drawing concept hierarchies use that method (e.g. ANACONDA, *Cernato*, *Concept Explorer*, *Elba*).

M1.5 Checking a Line Diagram of a Concept Hierarchy: A line diagram represents the concept hierarchy of a given finite context correctly if and only if the line diagram satisfies the following conditions: (1) each circle being the start of exactly one downward line segment must have attached an object name; (2) each circle being the start of exactly one upward line segment must have attached an attribute name; (3) an object g has an attribute m in the given context if and only if the names of g and m are attached to the same circle or there is an upward path of line segments from the circle with the name of g to the circle with the name of m; (4) the line diagram represents a concept hierarchy (cf. [GW99a], p.20, The Basic Theorem on Concept Lattices).

For line diagrams representing less than 50 concepts, it is quite easy to check the conditions (1), (2), and (3). Checking condition (4) by inspection is usually more costly because (4) mathematically means that the line diagram must represent a lattice. Nevertheless, experiences with many realistic data contexts have shown that a failure of condition (4) is usually accompanied by a failure of at least one of the conditions (1), (2), (3).

M1.6 Dualizing a Concept Hierarchy: *Dualizing a Context* means to interchange the roles of objects and attributes, i.e., objects become attributes and attributes become objects, while the context relation turns to its inverse. If one considers objects as instances of Firstness and attributes as instances of Secondness in the sense of Peirce's universal categories, dualizing a context can be understood as interchanging the roles of Firstness and Secondness. *Dualizing a Concept Hierarchy* means to interchange extension and intension in each concept, i.e., each concept becomes a concept of the dualized context, while each subconcept-superconcept-relationship turns to its inverse (cf. [GW99a], p.22). Therewith a line diagram of the dualized concept hierarchy can be obtained by turning a line diagram of the given concept hierarchy upside down.

There are contexts for which its dual context is meaningful, i.e., it is also interesting to view the attributes as objects and the objects as attributes. An example for that is the context in [Wi05a] on p.11 having the tones of the *diatonic scale* as objects and the major and minor triads as attributes; moreover, a tone as object has a triad as attribute if the tone belongs to the triad. This context might be understood as an answer to the questions: Which triads contain a given tone x in the diatonic scale? The dual context, in which the triads are the objects and the tones are the attributes, might be viewed as an answer to the questions: Which tones characterize a given triad y in the diatonic scale?

2.2 Determination of Concepts and Contexts

The basic mean for generating concepts of a given context are the two *derivations* assigning to each collection of objects the collection of all attributes which apply to those objects and assigning to each collection of attributes the collection of all objects which have those attributes. For determining concepts, sometimes even a context with its objects and attributes has to be determined from more general ideas.

M2.1 Generating Concepts: In a context, each collection of objects *generates a concept* the intension of which is the derivation of the given object collection and the extension of which is the derivation of that intension; dually, each collection of attributes *generates a concept* the extension of which is the derivation of the given attribute collection and the intension of which is the derivation of that extension (cf. [GW99a], p.18f.). An *object concept* is a concept generated by one object and an *attribute concept* is a concept generated by one attribute.

The extension of the concept generated by a given object collection is the *smallest concept extension* containing the generating object collection in the underlying context; this has as consequence that the intersection of concept extensions is an extension again. Dually, the intension of the concept generated by a given attribute collection is the *smallest concept intension* containing the generating attribute collection in the underlying context; this has as consequence that the intersection of concept intensions is an intension again.

M2.2 Generating All Concepts Within a Line Diagram: For smaller contexts the following procedure has been proven a success: First, represent the concept having the full object set of the given context as extension by a small circle and attach (from above) to that circle the names of all attributes which apply to all objects of the context. Secondly, choose from the left attributes all those the extension of which are maximal, draw for each of them a circle below the first circle, link them to the first circle by a line segment, and attach (from above) to them the corresponding attribute names. Then determine all intersections of the extensions of the already represented concepts and represent the concepts generated by those intersections by small circles with their respective line segments representing the subconcept-superconcept-relationships. Perform analogously the next steps until all attributes are treated. Finally, attach each object name (from below) to that circle from which upward paths of line segments lead exactly to those circles with attached names of attributes applying to the respective object (cf. [GW99a], p.64ff.).

After finishing the procedure, the user is recommended to check the representation by method M1.5. Quite often one has not represented all concepts; but usually it is not difficult to insert the missing circles and respective line segments. Finally, one should try to improve the drawn line diagram to obtain a better readable diagram.

M2.3 Determining All Concepts of a Context: A fast procedure for determining all concepts of a finite context is given by the so-called *Ganter Algorithm.* This algorithm is based on a lexicographic order on all collections of objects of the present context. For establishing this order, we assume a linear order $g_1, g_2, \ldots, g_n$ on all objects. Then an object collection A is defined to be *lectically smaller* than an object collection B if B contains the object which has the smallest index under all objects distinguishing A and B. The algorithm starts with the smallest concept extension, i.e., the derivation of the collection of all attributes, and continues by determining always the lectically next concept extension A^+ after the just determined extension A. The extension A^+ is generated by g_i and the object collection $\underline{A}$ consisting of all $g_1, \ldots, g_{i-1}$ contained also in A where i is the largest index for which g_i is not in A and the extension generated by $\underline{A}$ and g_i contains the same objects out of $g_1, \ldots, g_{i-1}$ as the extension A. The algorithm stops when it reaches the extension consisting of all objects (cf. [GW99a], p.66ff.). Finally, the constructed extensions are turned into concepts by M2.1. The subconcept-superconcept-relation can now be easily determined because it agrees with the containment relation between the concept extensions.

There are several *implementations* of the Ganter Algorithm (e.g. *ConImp* [Bu00], ANACONDA [Na96], *ConExp* [Ye00]) which allow to compute even large concept hierarchies and yield the input for drawing programs too. For drawing well-readable line diagrams of concept hierarchies by hand, a concept list is useful which indicates for each concept its upper neighbours in the concept hierarchy; in particular, it can be used to apply the so-called *geometric method* (see [GW99a], 69ff.).

M2.4 Determining a Context from an Ordered Collection of Ideas: There are situations in which it is desirable to elaborate concepts from more general ideas. This has caused a method for constructing a context from a collection of ideas ordered with respect to their generality. For such an idea collection a *downward/upward refinement* is defined to be a subcollection of ideas which contains with each idea all more general/special ideas of that idea and with every two ideas an idea more special/general than those two ideas. A downward refinement is called *irreducible* if, with respect to a suitable upward refinement, it is maximal under all downward refinements having no idea in common with that upward refinement. Dually, an upward refinement is called *irreducible* if, with respect to a suitable downward refinement, it is maximal under all upward refinements having no idea in common with that downward refinement. Now, for the desired context, we take the irreducible downward refinements as objects, the irreducible upward refinements as attributes, and the pairs of a downward and an upward refinement having some ideas in common as the object-attribute-relationships (cf. [SW86]).

An interesting application of the method M2.4 is the conceptual analysis of Aristotle's conception of the time continuum (cf. [Wi04b], p.460ff.). There the basic ideas are the time durations, which do not consist of time points. Time points can be derived as concepts generated by irreducible downward refinements of durations in the respective context constructed as above. More elementary constructions of contexts are discussed in the next subsection.

2.3 Conceptual Scaling

Scaling is the development of formal patterns and their use for analyzing empirical data. In *Conceptual Scaling* these formal patterns consists of contexts and their concept hierarchies which have a clear structure and reflect some meaning. Such a context is said to be a *conceptual scale* and its objects and attributes are called *scale values* and *scale attributes*, respectively (cf. [GW89], p.142ff.).

M3.1 Conceptual Scaling of a Context: A context may be connected with a conceptual scale by a *scale measure* which assigns to each object of the context a scale value in such a way that the collection of all objects assigned to values in any fixed extension of the conceptual scale is an extension too, which is called the *preimage* of the fixed scale extension under the considered scale measure. A system of scale measures on a logical context with values in respective conceptual scales is said to be a *full conceptual scaling* if every extension of the context is the intersection of the preimages of some scale extensions under the respective scale measures (cf. [GW89]).

The context of a *repertory grid test of an anorectic patient* discussed in [Wi00b] on p.365 permits a full conceptual scaling into three one-dimensional ordinal scales having the values 0, 1, 2 (cf. M3.4). The three scale measures assign to the object MYSELF the scale values 0,0,2, to IDEAL 0,1,1, to FATHER 1,0,0, to MOTHER 2,0,0, to SISTER 1,0,1, and to BROTHER-IN-LAW 0,2,0. The described full conceptual scaling leads to a well-readable diagram of the corresponding concept hierarchy in a 3-dimensional grid.

M3.2 Conceptual Scaling of a Many-valued Context: In a (complete) many-valued context every many-valued attribute assigns to an object a unique attribute value. Therefore, for turning a many-valued context into a context to obtain a related concept hierarchy, it is natural to interpret the many-valued attributes as scale measures and the attribute values as scale values of suitable conceptual scales (cf. M3.1). This motivates *conceptual scaling* of a many-valued context by which a (meaningful) conceptual scale is assigned to each many-valued attribute so that the corresponding attribute values are objects of that scale. The *derived context* of such conceptual scaling has the same objects as the many-valued context and has as its attributes the attributes of all assigned conceptual scales. An object of the derived context has an attribute of a specific scale if, in that scale, the attribute applies to the scale value which the respective many-valued attribute assigns to the object in the many-valued context (cf. [GW99a], p.36ff.).

The specific scaling methods which are mostly used are listed below. Further scaling methods are described in Section 1.3 and 1.4 of [GW99a].

M3.3 Nominal Scaling of a Many-valued Context: A context is called a *nominal scale* if each of its objects has exactly one attribute and each of its attributes applies to exactly one object. A conceptual scaling of a many-valued context is said to be *nominal* if all conceptual scales of the scaling are nominal (cf. [GW99a], p.42).

A nominally scaled many-valued context, having the former presidents of the Federal Republic of Germany as objects, is discussed in [Wi00b]. Its many-valued attributes are the *age of entrance* with the values < 60 and > 60, the *terms of office* with the values 1 and 2, and the *party* with the values CDU, SPD, and FDP. Therefore, the derived context has the seven attributes *age of entrance:* < 60, *age of entrance:* > 60, *terms of office:* 1, *terms of office:* 2, *party:* CDU, *party:* SPD, and *party:* FDP. Each president has three attributes, namely the value of his age of entrance, of his terms of office, and of his party, respectively.

M3.4 Ordinal Scaling of a Many-valued Context: A context is called an *ordinal scale* if its objects and its attributes carry hierarchical order relations which are in one-to-one correspondence and if an object has an attribute exactly in case the object is in the order relation with the object corresponding to the attribute (or, equivalently, in case the attribute is in the opposite order relation with the attribute corresponding to the object). An ordinal scale is *one-dimensional* if the objects and attributes with their hierarchical order relations form corresponding increasing chains. A conceptual scaling of a many-valued context is said to be (*one-dimensional*) *ordinal* if all conceptual scales of the scaling are (one-dimensional) ordinal (cf. [GW99a], p.48 and p.42).

An ordinally scaled many-valued context, having 26 places along the Canadian Coast of Lake Ontario as objects, is discussed in [SW92]. Its many-valued attributes are five *tests concerning water pollution*, the attribute values of which are six *segments of potential measurement values*, respectively. Therefore, the derived context has 30 attributes: each of the 5 tests combined with one of its 6 segments. A place has a segment of a test as its attribute if the measurement value of the test at this place lies in the segment or is larger than all values of that segment. Clearly, each test represents a one-dimensional ordinal scale.

M3.5 Interordinal Scaling of a Many-valued Context: A context is called a (one-dimensional) *interordinal scale* if it is the juxtaposition of a (one-dimensional) ordinal scale and its opposite scale, i.e., an object has an attribute exactly in case the object is in the opposite order relation with the object corresponding to the attribute. A conceptual scaling of a many-valued context is said to be (*one-dimensional*) *interordinal* if all conceptual scales of the scaling are (one-dimensional) interordinal (cf. [GW99a], p.57 and p.42).

In the retrieval system developed for the library of the center of interdisciplinary technology research at Darmstadt University of Technology (see M6.3), a one-dimensional interordinal scale is used to represent time periods in which the books of the library have been published. The chosen objects of that scale are the time periods *before 1945*, *1945-1959*, *1960-1969*, *1970-1979*, *1980-1984*, *1985-1989*, *1990-1993*, *from 1994* and the scale attributes are the time periods *before 1945*, *before 1960*, *before 1970*, *before 1980*, *before 1985*, *before 1990*, *before 1994*, and *from 1945*, *from 1960*, *from 1970*, *from 1980*, *from 1985*, *from 1990*, *from 1994*. Naturally, a time period object is considered to have a time period attribute if the object period is contained in the attribute period. The concept hierarchy of the defined interordinal scale has a well-readable line diagram which is shown in [RW00] on p.250.

M3.6 Contraordinal Scaling of a Many-valued Context: A logical context is called a *contraordinal scale* if its objects and its attributes carry hierarchical order relations which are in one-to-one correspondence so that an object has an attribute exactly in case the object is not in the opposite order relation with the object corresponding to the attribute (or, equivalently, in case the attribute is not in the order relation with the attribute corresponding to the object). A conceptual scaling of a many-valued context is said to be *contraordinal* if all conceptual scales of the scaling are contraordinal (cf. [GW99a], p.49). The special case that the order relations are just the equality relations yields the so-called *contranominal scales* in which all subcollections of objects are extensions and all subcollections of attributes are intensions (cf. [GW99a], p.48).

The ordinally scaled many-valued context and its corresponding concept hierarchy presented in [GW99a] on p.44/45 reports ratings of sights on the Forum Romanum in Rome taken from the travel guides *Baedecker* (*B*), *Les Guides Bleus* (*GB*), *Michelin* (*M*), and *Polyglott* (*P*). The four-dimensional structure caused by the four guides could be made more transparent by a contraordinal scaling of the many-valued context as shown in [Wi87] on p.196. This scaling yields a derived context with the seven attributes [no star in B], [no star in GB], [no or one star in GB], [no star in M], [no or one star in M], [no or one or two stars in M], and [no star in P].

M3.7 Convex-Ordinal Scaling of a Many-valued Context: A logical context is called a *convex-ordinal scale* if it is the juxtaposition of a contraordinal scale and its opposite scale, i.e., a scale in which an object has an attribute exactly in case the object is in the opposite negated order relation with the object corresponding to the attribute. A conceptual scaling of a many-valued context is said to be *convex-ordinal* if all conceptual scales of the scaling are convex-ordinal (cf. [GW99a], p.52).

Convex-ordinal scales are often derived from hierarchically ordered structures. Such a structure is, for instance, presented in [SW93] in Figure 1 by a diagram representing 35 dyslexics ordered by their numerical scores obtained from three tests. The ordering locates a person below another one if her scores do not exceed the corresponding scores of the other person, but at least one score is even less the corresponding score of the other person. For dissecting the 35 dyslexics into widely uniform training groups it is desirable that each person located by the ordering between two persons of a group should also belong to that group. This rule has as consequence that the groups are extensions of the convex-ordinal scale canonically derivable from the described ordering (cf. [GW99a], p.52).

2.4 Conceptual Classification

Classifying objects is an important activity of human thinking which is basic for interpreting realities. There is a wide spectrum of methods to perform classifications and the interest is even to develop further methods. Especially, there is a strong demand for *mathematical methods of classification* which can be implemented on computers. This has stimulated a rich development of numerical classification methods which are of extensive use today. But those methods are also criticized because of a major limitation, in that the resulting classes may not be well characterized in some human-comprehensible language (cf. [SW93]). *Conceptual Classification*, which uses concept hierarchies of contexts, overcomes this limitation by incorporating a conceptual language based on attributes and attribute values.

M4.1 Concept Classification of Objects: The first step of *conceptually classifying objects* is to choose appropriate attributes according to the purpose of the approached classification. Then the logical context for the considered objects and attributes has to be established and after that its concept hierarchy. This hierarchy yields the desired classification, the object classes of which are just the non-empty extensions of the concepts forming the hierarchy.

An example of a concept classification is the logical support of the educational film *"Living Being and Water"* mentioned in [GW99a] on p.18 and p.24. This film was produced by the Institute of Educational Technology in Veszprém/Hungary. For developing the film, the first decision was to emphasize on the general objects leech, bream, frog, dog, spike-weed, reed, bean, and maize as well as on nine attributes from "needs water to live" to "suckle its offsprings". After determining the respective context, its concept hierarchy with its object classification was derived which supported not only the design and production of the film, but also the evaluation of its perception.

It has to be mentioned that, in our example, the object extensions do not form a tree as is often required for classifications (see e.g. [RS84]). In the German Standard DIN 2331 from 1976 about concept systems, classifications being tree-like are called *monohierarchical systems*, otherwise *polyhierarchical systems*; in this way the standard respects that classifications in practice are quite often not trees.

M4.2 Many-valued Classification of Objects: *Conceptually classifying objects* of many-valued contexts presupposes a conceptual scaling, the specific method of which is appropriately chosen according to the purpose of the approached classification, respectively. The *derived context* of such conceptual scaling has already been described in method M3.2. The special quality of a many-valued classification is that each many-valued attribute can function as a semantic criterion for which the attributes of the respective scale represent meanings specifying the criterion.

An example of a many-valued classification is the logical support of an *investigation in developmental psychology* (cf. [SW93]). The data of that investigation have been concentrated in a many-valued context, the objects of which are 62 children from the age of 5 to 13 and the attributes of which are 9 general criteria of concept development and the attribute age. The investigation was performed with the aim to reconstruct the developmental sequences of the concept *work*. The analysis of those sequences was based on the criteria *quality of motives*, *generalization*, and *structural differentiations* which give the most differentiated view of changes and advances in development. The many-valued context was convex-ordinally scaled by method M3.7 to a logical context so that the children could be classified in seven meaningful extensions representing levels of concept development. The most interesting result was that some children reached earlier a higher level of generalization than others who reached earlier a higher level of quality of motives. Such a kind of branching in concept development has not been proven before.

2.5 Analysis of Concept Hierarchies

The term "Analysis" means an investigation by dissecting a whole into suitable parts to obtain a better understanding of the whole. Thus, *analyzing a concept hierarchy* consists in partitioning its concepts into meaningful parts which together form a subdivided conceptual structure leading to an improved understanding of the concept hierarchy.

M5.1 Partitioning the Attributes of a Context (Nested Line Diagram): For studying larger concept hierarchies of contexts it has been proven useful to *partition the attributes* of the given context in classes and to identify the subcontexts formed by one of those classes and the objects of the whole context, respectively. Then, each concept of the whole context is represented by a *sequence of subcontext concepts*, just one from each identified subcontext; the intension of such a subcontext concept consists of all attributes of the represented concept which are also attributes of the subcontext concept. The resulting subdivided structure of the whole concept hierarchy can be visualized by a *nested line diagram* constructed as follows (cf. [GW99a], p.75ff.): First, line diagrams of the concept lattices of the subcontexts are prepared and ordered in a sequence of the same kind as the corresponding subcontexts. Then, the line diagram being second in the sequence is copied into each circle of the line diagram being first in the sequence; next, the line diagram being third in the sequence is copied into each circle of each copy of the line diagram being second in the sequence; and so on, until the line diagram being last in the sequence is copied into each circle of each copy of the line diagram being last but one in the sequence. Finally, each concept of the whole context and its sequence of subcontext concepts is indicated by a corresponding sequence of circles each of which contains the next one and represents the corresponding subcontext concept; such sequence of circles can be marked by only distinguishing the last circle of the sequence. This method is used, in particular, for applying the TOSCANA-aggregation (s. M6.3).

A well-readable nested line diagram of a concept hierarchy with 139 concepts concerning *old Chinese urns* is presented in [Wi84] on p.42. The underlying context has eight pairs of dichotomic attributes from which five pairs form a one-dimensional interordinal scale (cf. M3.5). For the attribute partition, those ten attributes were taken as the first attribute class, two further pairs as the second attribute class, and the last pair as the third attribute class. The concept hierarchies of the corresponding subcontexts consist of 22, 10, and 4 concepts, respectively. The subdivided structure diagram underlying the nested diagram has 880 very small circles, 220 small circles containing 4 very small circles, and 22 larger circles containing 10 small circles, respectively. Since the smallest concepts of the three hierarchies have an empty extension, one can erase all circles representing a concept with an empty extension (except the lowest very small circle) so that structure diagram consist of only 661 very small circles.

M5.2 Atlas-Decomposition of a Concept Hierarchy: Analyzing larger concept hierarchies may be stimulated by the atlas metaphor. Such approach can be based on the notion of a *block relation* which relates objects and attributes, in particular, if the object has the attribute in the underlying context; furthermore, the block relation derivation of each object/attribute is an extension/intension of the original context (cf. [GW99a], p.121ff.). This guarantees that each concept of the block relation context gives rise to an interval in the concept hierarchy of the original context consisting of all concepts the extension/intension of which is contained in the extension/intension of the block relation concept. Metaphorically, those intervals are the maps of the respective atlas. As in an atlas, for many applications it is desirable that neighbouring maps overlap.

A meaningful atlas-decomposition of a concept hierarchy concerning the *harmonic forms of the diatonic scale* is discussed in [Wi84] on p.45ff. For that hierarchy the used block relation yields the largest decomposition with overlapping neighbouring maps. Each map clarifies the relationships between harmonic forms differing just by one tone.

M5.3 Concept Patterns in a Concept Hierarchy: Concept hierarchies may be understood as source of well-interpretable *concept patterns*, for instance, as concept chains, ladders, trees, grids etc. Such patterns are considered as *conceptual measurement structures* as discussed in [GW99a] in section 7.3 and 7.4. A specific method of identifying concept patterns is based on the search of respective *subcontexts* constituted by suitable objects and attributes of the underlying context (cf. [GW99a], section 3.1). Such search is quite often successful if one tries to find long sequences of attributes, the extensions of which form a chain with respect to containment, and completes those attributes to a subcontext having enough objects to represent those chains.

A meaningful example about the support of designing working places for handicapped people is discussed in [Wi87], p.188ff. The established logical context indicates which part of the human body is affected by which demand of work. A well-drawn line diagram of the respective concept lattice shows a dominant two-dimensional grid pattern which is generated by two sequences of attributes concerned with body movements, one from climbing over waking and squating to foot moving and the other from climbing over reaching and holding to seizing. Another instructive example is the concept hierarchy in [Wi92] about the colour perception of a gold fish. This hierarchy which consists of 141 concepts could only be well-drawn and well-interpreted because of the discovery of two long attribute chains representing parts of the colour circle.

2.6 Aggregation of Contexts and Concept Hierarchies

Knowledge is often represented not in one, but in several contexts. Clearly, it is desirable to aggregate those contexts to a common context, so that the single contexts are derivable as direct as possibly from the common context. Furthermore, the construction of the concept hierarchy of the common context by the concept hierarchies of the single contexts should be known, as well as the projections from the common concept hierarchy onto the single concept hierarchies, respectively.

M6.1 Juxtaposition of Contexts with Common Object Collection: Forming the *juxtaposition of given contexts with common object collection* means to establish the context having as objects those of the common object collection and as attributes those which are attributes of one of the given contexts (attributes from different contexts are considered to be different too); in the juxtaposition, an object has an attribute if it has the attribute in the context containing that attribute (cf. [GW99a], p.40). The *concepts of the juxtaposition* are generated by the intersections of the extensions of concepts of the single contexts (cf. M2.1). Conversely, each *concept of a single context* is the projection of all concepts of the juxtaposition having the same extension as that concept.

An extensive project of Conceptual Knowledge Processing highly dependent on the juxtaposition aggregation was the development of an *information system about laws and regulations concerning building constructions* requested by the Department for Building and Housing of the State Nordrhein-Westfalen. The necessary knowledge for that project was represented in contexts and their concept hierarchies concerned with specific themes such as "fundamental construction of a family house" [EKSW00], "functional rooms in a hospital" [Wi05b], "operation and fire security" [KSVW94] etc. The concept hierarchy of the juxtaposition of the mentioned hospital and security context represented by a nested diagram can also be found in [KSVW94] (the common object collection was formed by all relevant information units about laws and regulations concerning building constructions).

M6.2 Aggregation Based on Object Families: A general framework for this method is the so-called "semiproduct of contexts" (cf. [GW99a], p.46). The *semiproduct* of a collection of contexts is a context the objects of which are all object families having exactly one object from each context of the collection and the attributes of which are just the attributes of the given contexts (attributes from different contexts are viewed to be different); in the semiproduct, an object family has an attribute if that attribute applies in its respective context to the unique object belonging to the respective context and to the considered object family. An *Aggregation Based on Object Families* is a subcontext of a semiproduct of contexts having meaningful object families as its objects, while its attributes are just all attributes of the semiproduct.

The method "Aggregation Based on Object Families" plays an important role in [BS97] (cf. also [GW99b]); in particular, applications of this method to switching network are suggested. Such applications and their theoretical background have been elaborated in [Kr99], where the method "Aggregation Based on Object Families" is, for instance, used to analyse a *lighting circuit with emergency light*. The analysis yields four contexts: one for the main switch with three states, another one for a switch with two states linking either to the main-light or to the emergency light, and two further contexts with two states indicating whether the main light or the emergency light is on or out, respectively. The semiproduct of the four contexts has as objects the $24(= 3 \cdot 2 \cdot 2 \cdot 2)$ quadruples of states and as attributes the $7(= 3 + 2 + 1 + 1)$ attributes of the four contexts. Only 6 of the 24 quadruples are meaningful (i.e. they satisfy the so-called network rule); hence a 6×7-subcontext of the semiproduct represents the logic of the analysed lighting circuit with emergency light.

M6.3 TOSCANA-Aggregation of Concept Hierarchies: The idea of a TOSCANA-aggregation is to view a related system of concept hierarchies metaphorically as a *conceptual landscape of knowledge* [Wi97b] which can be explored by a purpose-oriented combination and inspection of suitable selections of the given concept hierarchies. This is logically supported by line diagrams representing concept hierarchies of juxtapositions of contexts in the sense of M6.1 and suitable restrictions of those hierarchies (cf. [KSVW94], [EKSW00]). For performing the TOSCANA-aggregation method, software has been developed since 1990; the most advanced software is available by the programs of the ToscanaJ *Suite* [BH05] which are developed as Open Source project on Sourceforge (http://sourceforge/projects/toscanaj).

The development of the TOSCANA-aggregation method was stimulated by a research project with political scientists in the late 1980th. The task was to analyse a data context with 18 objects, namely norm- and rule-guided international cooperations, so-called *regimes*, and 24 many-valued attributes representing factors of influence, typological properties, and regime impacts (cf. [KV00]). This many-valued context was conceptually scaled by the method M3.2, where the used scales arose as result of an interdisziplinary co-operation between the coworking mathematicians and political scientists. Then more then fifty subcontexts of the scaled many-valued context together with the corresponding concept hierarchies were produced by hand for answering special research questions. This made clear that a mathematically founded construction method and its implementation would be desirable which allows the aggregation of arbitrarily chosen conceptual scales. The TOSCANA method and software met this desire and gave rise to the development of many TOSCANA-systems in a wide spectrum of disciplines. In particular, the research on international regimes has benefited from this development which is witnessed by a new TOSCANA-system about 90 regime components the data of which were elaborated by a great number of international political scientists over more than four years (cf. [Ks05]).

2.7 Conceptual Identification

A method is considered to be a *conceptual identification* if it determines concepts which classifies given instances. A well-known example of a conceptual identification is to determine the position of an individual plant in a taxonomy of plants.

M7.1 Identifying a Concept: The elementary type of conceptual identification is the *classification of an instance* by a given system of concepts. Methodollogically, such identification can be well performed on the basis of a context, the objects of which are the classes of the given classification system; the attributes of the context are used for the identification process that increasingly determines those attributes which apply to the considered instances. It is advantageous to visualize this process in a line diagram of the concept hierarchy of the context by indicating the decreasing path from concept to concept generated by the determined attributes until no further attribute which apply to the considered instance leads to a new subconcept. Then the concept generated by all determined attributes is the identified concept for the considered instance (cf. [KW86]).

The described identification process can be effectively represented and supported by a computer. That has, for instance, been done for identifying the symmetry types of two-dimensional patterns. In this case, the computer screen shows the user a line diagram representing the concept hierarchy having as objects the considered symmetry classes. For a given two-dimensional symmetry pattern, the user tries to find enough attributes which apply to the given pattern. After each input of such an attribute, the screen highlights the concept which is generated by the already fed attributes. This process may, for instance, identify the symmetry type the concept of which is generated by the attributes "admitting two reflections with non-parallel axes", "admitting a rotation of 90°", and "admitting a rotation of 90° the center of which is not on a reflection axis" (see [Wi00b], p.362f.). A well-designed computer implementation for identifying the symmetry types of two-dimensional patterns has been successfully offered to the visitors of the large Symmetry Exhibition at the Institute Mathildenhöhe in Darmstadt 1986 (cf. [Wi87], p.183ff.).

M7.2 Identifying Concept Patterns: There are many families of contexts the concept hierarchies of which offer a specific type of *regular concept patterns* for interpreting conceptual structures; in particular, the so-called *standard scales* (discussed in [GW99a], Section 1.4) are such concept patterns. In a context, a concept pattern is *identified* by a collection of objects of the context if each concept of the pattern is generated by a subcollection of those objects; it is *strongly identified* if, in addition, each subcollection of the objects generates a concept of the pattern.

How the identification of concept patterns may support the interpretation of empirical data shall be briefly demonstrated by an example from linguistics. In the late 1980th, the dialectician H. Goebl from the University Salzburg became interested in the application of Formal Concept Analysis to his empirical data; in particular, he offered the Darmstadt research group data about *phonemes of French in Swizerland*. Those data were transformed in a context having 63 measurements points as objects and 40 phonetic characteristics as attributes. Since it is of special interest to study modifications of the phonemes along measurement points, one-dimensional interordinal scales (cf. M3.5) yield well-interpretable concept patterns. Indeed, the multifarious modifications of phonemes could be shown by the concept hierarchies of such scales strongly identified by 3, 4, and 5 consecutive measurement points, respectively (see [FW89]).

2.8 Conceptual Knowledge Inferences

Conceptual Knowledge Processing does not only rely on the representation of conceptual structures, but also on *conceptual inferences* which are inherent in knowledge structures. The importance of inferences for human thinking has been, in particular, underlined by R. Brandom in his influential book “Making it explicit. Reasoning, representing, and discursive commitment” [Br94]. According to Brandom, knowledge is founded on an *inferential semantics* which rests on material inferences based on a normative pragmatics. In Formal Concept Analysis, up to now, the research on inferences has been dominantly concentrated on implications and dependences (cf. [GW99a], Section 2.3 and 2.4).

M8.1 Determining the Attribute Implications of a Context: In a given context, the attributes $m_1, \ldots, m_k$ *imply* the attributes $n_1, \ldots, n_l$ if each object having the attributes $m_1, \ldots, m_k$ also has the attributes $n_1, \ldots, n_l$. Such implication can be determined within a line diagram of the concept hierarchy of the given context as follows: First one identifies the circle representing the largest common subconcept $\mathfrak{c}$ of the attribute concepts generated by the attributes $m_1, \ldots, m_k$, respectively; then the attributes implied by $m_1, \ldots, m_k$ are recognizable as those attributes n which generate a superconcept of $\mathfrak{c}$, i.e. there is an ascending path from the circle representing $\mathfrak{c}$ to the circle representing the attribute concept generated by n.

There is a number of *implemented algorithms* for determining bases of attribute implications, the mostly used algorithm of which has been developed by B. Ganter [Ga87] (see also [GW99a], Section 2.3). The Ganter algorithm determines the *stem basis* for all attribute implications of a given context (also called the Duquenne-Guigues-Basis [GD86]); all other bases can be easily derived from the stem basis. Applying the stem basis shall be briefly demonstrated by the analysis of properties of drive concepts for motorcars presented in [Wi87], p.174ff.: The context for this analysis has as objects the drive concepts “Conventional”, “Front-wheel”, “Rear-wheel”, “Mid-engine”, and “All-wheel” and as attributes 25 properties such as “good road holding”, “under-steering”, “high cost of construction” etc. The Ganter algorithm yields 31 attribute implications as, for instance, “good economy of space” implies “good road holding”, “bad economy of space” implies “low cost of construction”, “good drive efficiency unloaded” and “good maintainability” imply “low cost of construction”. The few cited implications may already show how valuable the stem basis can be for the interpretation of the given data context.

M8.2 Determining Many-valued Attribute Dependencies: Dependencies between many-valued attributes are of great interest in many fields of empirical research. A basic type of such dependencies are the functional dependencies: In a (complete) many-valued context, many-valued attributes $n_1, \ldots, n_l$ are *functionally dependent* on many-valued attributes $m_1, \ldots, m_k$ if, for every two objects g and h of the many-valued context, the corresponding attribute values $m_i(g)$ and $m_i(h)$ $(i = 1, \ldots, k)$ are equal then the corresponding attribute values $n_i(g)$ and $n_i(h)$ $(i = 1, \ldots, l)$ are equal too. For determining functional dependencies, a method has been proven useful which is based on the following context derived from the given many-valued context: the *derived context* has as objects all pairs of two (different) objects of the many-valued context and as attributes all its many-valued attributes where a pair of objects g and h is related to a many-valued attribute m if the corresponding attribute values $m(g)$ and $m(h)$ are equal. Then it can be proved that many-valued attributes $n_1, \ldots, n_l$ are *functionally dependent* on many-valued attributes $m_1, \ldots, m_k$ in the many-valued context exactly if $m_1, \ldots, m_k$ implies $n_1, \ldots, n_l$ in the derived logical context. This equivalence allows us now to use method M8.1 to determine all functional dependences of the given many-valued context. If one replaces equality by the inequality $\leq$, respectively, one gets the analogous result for *ordinal dependency* (cf.[GW99a], p.91f.).

A prominent field for applying functional dependencies is Database Theory. Ordinal dependencies have been successfully applied in Measurement Theory. For instance, in [WW96], it is shown that enough ordinal dependencies in an ordinal many-valued context guarantees a linear representation of the context in a vector space over the field of real power series. This is demonstrated by a two-dimensional representation of a data context about the colour perception of a goldfish (cf. also [WW04]).

2.9 Conceptual Knowledge Acquisition

The *central idea of knowledge acquisition* in the frame of Formal Concept Analysis lies in the assumption that, in the field of exploration, the conceptual knowledge can be thought to be represented by a context with finitely many attributes and by its concept hierarchy; such a context is called a *universe*. The exploration of knowledge starts with some partial information about the considered universe and acquires more information by phrasing questions which are answered by experts. For this procedure it is a main concern not to ask questions which can already be answered by the acquired knowledge (cf. [Wi89b]).

M9.1 Attribute Exploration: For an attribute exploration, first a *universe* is specified as a context the attribute of which are explicitly given, but the objects of which are only known to belong to a certain type of objects. It is often helpful to choose at the beginning of the exploration some objects and to determine the subcontext of the universe based on those objects together with the attributes of the universe. Then an *implementation of the algorithm* described in [GW99a], p.85, should be used which leads to questions whether certain attribute implications are valid in the universe or not. If yes, then the actual attribute implication is added to the list of already recognized valid attribute implications of the universe. If not, then an object of the universe has to be made explicit which has all attributes of the premise of the actual implication, but has not at least one attribute of its conclusion; such a new object is used to extend the actual explicit subcontext to a new subcontext of the universe. Since the universe has only finitely many attributes, the exploration ends after finitely many steps. Then the resulting subcontext has the same concept intensions as the assumed universe.

In [GW99a], p.86ff., the *attribute exploration* is demonstrated by the universe which has as objects all binary relations between natural numbers and as attributes the properties “reflexive”, “irreflexive”, “symmetric”, “asymmetric”, “antisymmetric”, “transitive”, “negatively transitive”, “connex”, and “strictly connex”. The concept hierarchy of the resulting subcontext consists of 50 concepts the structure of which clarifies completely the *implication logic* of the given nine properties of binary relations (cf. M8.1).

M9.2 Concept Exploration: For a concept exploration, first a *universe* is specified as a context the objects and attributes of which are only known to belong to a type of objects and a type of attributes, respectively; in addition, a finite number of concepts of the universe are specified by their names. Then the *aim of the concept exploration* is to identify all concepts of the universe which can be deduced from the specified concepts by iteratively forming the largest common subconcept and the smallest common superconcepts of already constructed concepts in the universe. This procedure is accompanied by the questions whether for two concepts one is a subconcept of the other or not. If yes, then this order relationship is added to the table of already recognized pairs of subconcept-superconcept of the universe. If not, then an object belonging to one concept and an attribute belonging to the other concept and not applying to the object have to be made explicit. The acquired knowledge, as it is accumulating, is represented in a context which has as objects the explicitly made objects together with the constructed concepts and as attributes the explicitly made attributes with the constructed concepts too; the context relation indicates which explicit object has which explicit attribute, which explicit object belongs to which constructed concept, which explicit attribute belongs to which constructed concept, and which constructed concept is the subconcept of another constructed concept (cf. [Wi89b], p.375ff.).

The concept exploration is demonstrated in [Wi89b] within the universe having as objects all countable relational structures with one binary relation R and as attributes all universal sentences of first order logic with the relational symbol 'R' and equality; a relational structure as object has a sentence as attribute if the structure satisfies the sentence. In particular, the attribute exploration is explicitly performed with the three specified concepts "orthogonality", "dominance", and "covering". The acquired knowledge is represented in a 14×14-context based on 4 objects, 4 attributes and 10 concepts. More examples and theoretical developments can be found in [St97].

M9.3 Discovering Association Rules: In a context, an *association rule* is an ordered pair $(X \to Y)$ of attribute collections X and Y for which the following relative frequencies are computed: the *support* of $(X \to Y)$ is the number of attributes which are in X or Y divided by the number of objects in the context, and the *confidence* of $(X \to Y)$ is the number of attributes which are in X or Y divided by the number of attributes in X. The task is to determine all ordered pairs $(X \to Y)$ for which the support of $(X \to Y)$ is above a given *support threshold* chosen from the interval $[0, 1]$ and the confidence of $(X \to Y)$ is above a chosen *confidence threshold* chosen from the interval $[0, 1]$. In solving the task, the crucial part is the determination of all key-attribute-collections, which are attribute collections being minimal in generating a concept (cf. M2.1). The method of doing that is based on the observation that each subcollection of a key-attribute-collection is also a key-attribute-collection. Thus, an effective procedure can be designed which tests first the one-element attribute collections of being key-attribute-collections, then the two-element attribute collections not properly containing a key-attribute-collection and so on (for more details see [LS05]).

Association rules are, for instance, used in warehouse basket analysis, which is carried out to learn which products are frequently bought together. A general overview about discovering and applying association rules can be found in [LS05], Section 6.

2.10 Conceptual Knowledge Retrieval

Since *Information and Knowledge Retrieval* deals with organizing, searching, and mining information and knowledge, methods of Conceptual Knowledge Processing may support retrieval activities. They can do this by effectively complementing the existing search systems, in particular, by visualizing retrieval results, improving individual search strategies, and hosting multiple integrated search strategies (cf. [CR04], [CR05]).

M10.1 Retrieval with Contexts and Concept Hierarchies: The retrieval of documents can be seen to take place in a *context* the objects of which are the available documents and the attributes of which are the constituents of queries. Then the intension of a concept of such context contains all *queries* having as retrieval result exactly all documents in the extension of the considered concept. The concept hierarchy of the context shows especially which queries yield neighbour concepts causing only minimal changes between the retrieved extensions (cf. [GSJ86]). In general, a concept hierarchy of a context based on a number of retrieval results may give a useful overview which faster leads to fulfill the purpose of the search (cf. [CR05]).

In [Ko05], the method M10.1 is applied and elaborated to develop an improved *front end to the standard Google search.* The basic idea of this development is to use Google's three-row result itemset consisting of title, short description, and the uniform resource locators (URL) to build a context and its concept hierarchy. The context has as objects the first n URLs and as attributes the meaningful feature terms extracted from Google's first n three-row results (the number n is eligible). The context is presented best by a cross table in which the names of the attributes, applying to the same objects, are heading the same column (cf. M1.2). It turns out that already this presentation of the retrieved results is often very useful because a larger manifold of information units can be viewed at once and selectively compared. This effect can even be increased if the corresponding concept hierarchy is visualized.

M10.2 Retrieval with a TOSCANA-System: Conceptual knowledge retrieval is often a process in which humans search for something which they only vaguely imagine. Therefore humans organize such processes not only by a sequence of queries in advance, they also learn step by step how to specify further what they are actually searching for. Such interactive retrieval and learning process can be successfully supported by a suitable *TOSCANA-system* established by the TOSCANA-aggregation method (M6.3). The TOSCANA-system is structured by a multitude of conceptual scales (cf. Section 2.3) which are applied as search structures to the objects under considerations. The line diagrams of the activated scales are shown to the user who learns by inspecting them how to act further (cf. [BH05]).

Retrieval with a TOSCANA-system has been, for instance, established by developing a *retrieval system for the library of the "Center of Interdisciplinary Technology Research*" (ZIT) at the TU Darmstadt using the method M6.3 (cf. [RW00]). For supporting the search of literature, a related system of 137 concept hierarchies was developed. The underlying contexts of those hierarchies have as objects all books of the library and as attributes well-chosen catch words which represent a specific theme, respectively. In [Wi05b], p.17, there is a report on a literature search in the ZIT-library concerning the theme "expert systems dealing with traffic". This search starts with the concept hierarchy "Informatics and Knowledge Processing" which has "Expert Systems" as one of its attributes; the corresponding line diagram shows that there are 60 books in the library having "Expert Systems" as assigned catchword. This suggests to the user to consider the concept hierarchy "Town and Traffic" restricted to those 60 books; then the resulting line diagram shows that 9 of the 60 books have also "Traffic" as an assigned catchword and, additionally, 4 resp. 1 of those 9 books "Means of Transportation" resp. "Town" as assigned catchword.

2.11 Conceptual Theory Building

Empirical theory building, in particular in the human and social sciences, may be logically supported by methods of Conceptual Knowledge Processing. The basic models used by those methods are contexts and their concept hierarchies which allow the *representation of scientific theories* in a way that the theories become structurally transparent and communicable (cf. [SWW01]).

M11.1 Theory Building with Concept Hierarchies: Conceptual theory building starts from *data and information* which are mostly represented in data tables, texts, images, and inferential connections. The goal is to generate a rich, tightly woven, explanatory theory that closely approximates the reality it represents. Methodologically, this aims at a suitable representation of the considered data and information by a unifying concept hierarchy. That often leads to question the data and information and to work further with their improvements. Thus, *conceptual theory building* is an inductive process which stepwise improves theories which are always represented by concept hierarchies (cf. [SC90]).

Interesting examples of conceptual theory building are the development of everyday theories of logical relationships. There is a kind of surprising evidence that a great deal of those theories are determined by attribute implications with one-element premise and by incompatibilities between attributes (cf. [Wi04a]). For instance, the theory about the core of the *lexical field of waters* can be characterized by the dichotomic pairs of attributes “natural - artifical”, “running - stagnant”, “constant - temporary”, and “inland - maritime”. Applying those attributes to words describing the types of waters like “plash”, “channel”, “sea” etc. teaches that the types of waters satisfy the following attribute implications with one-element premise:

“temporary” ⇒ “natural”, “stagnant”, “inland”;
“running” ⇒ “constant”, “inland”;
“artificial” ⇒ “constant”, “inland”;
“maritime” ⇒ “natural”, “stagnant”, “constant”.

These implications together with the incompatibilities described by the four dichotomic attribute pairs completely determine the theory of implicational relationships in the considered core of the lexical field of waters (this has been shown in [Wi04a] by using the empirical results of [Kc79]). The concept hierarchy representing that theory is presented on the cover of [GW99a].

M11.2 Theory Building with TOSCANA: Conceptual theory building can be based on the method “TOSCANA-Aggregation of Concept Hierarchies” (M6.3) applied to an empirically derived collection of objects. For *building-up the aimed theory*, this object collection is structured by justified conceptual scales. Then interesting aggregations of those scales and their concept hierarchies are tested with regard to their meaningfulness concerning the approached theory. This testing might suggest improvements of the scales and their corresponding concept hierarchies which are then tested again. The goal is to reach a well-founded *TOSCANA-system* which adequately represents the aimed theory.

Theory building with TOSCANA has been substantially applied to support a dissertation about *"Simplicity. Reconstruction of a Conceptual Landscape in the Esthetics of Music of the 18th Century"* [Ma00]. The methodological foundation for this application was elaborated in [MW99]. The empirical collection of objects was given by 270 historical documents which were made accessible by a normed vocabulary of more then 400 text attributes. Those text attributes were used to form more general attributes for the conceptual scales of the approached TOSCANA-system. By repeatedly examining and improving aggregations of scales and their concept hierarchies, a well-founded TOSCANA-system was established which successfully supported the musicological research.

2.12 Contextual Logic

Contextual Logic has been introduced with the aim to support knowledge representation and knowledge processing. It is grounded on the traditional philosophical understanding of logic as the doctrine of the forms of human thinking. Therefore, Contextual Logic is developed by mathematizing the philosophical doctrines of concepts, judgments, and conclusions (cf. [Ka88], p.6). The mathematization of concepts follows the approach of *Formal Concept Analysis* [GW99a], and the mathematization of judgments uses, in addition, the *Theory of Conceptual Graphs* [So84]. The understanding of logic as the doctrine of the forms of human thinking has as consequence that main efforts are undertaken to investigate the mathematical and logical structures formed by (formal) concepts and concept(ual) graphs (cf. [Wi00a]).

M12.1 Conceptual Graphs Derived from Natural Language: A *conceptual graph* is a labelled graph that represents the literal meaning of a sentence or even a longer text. It shows the *concepts*, represented by boxes, and the *relations* between them, represented by ovals. The boxes contain always a name of a concept and, optionally, a name of an *object* belonging to that concept; no object name in the box means that there exist an object belonging to the concept named in the box. The ovals contain always a name of a relation which relates all the objects the names of which are contained in the boxes linked to the oval of that relation. In stead of repeating an object name in several boxes, it is allowed to write the name in only one box and to link this box to all those other boxes by broken lines (for more information see [So92]).

The representation by conceptual graphs has been practiced in many application projects concerning conceptual knowledge processing and has stimulated further useful theories (cf. contributions to the *International Conferences on Conceptual Structures* documented in the Springer Lecture Notes in Computer Science since 1993). One of such theories is the *Contextual Judgment Logic* the start of which was stimulated by a conceptual graph representation of a text about Seattle's central business district [Wi97a]. A quite special project was performed in a classroom of grade 6 with 32 boys and girls to clarify the question: Can already young pupils be trained in the ability of formal abstraction by transforming natural language into conceptual graphs? It turned out that most of the pupils learned very fast to turn simple sentences into a graphically presented conceptual graph. Already in the third lesson they were able to glue rectangular and oval pieces of paper on a cardboard in a way that they could inscribe and link those pieces to represent a little story by a conceptual graph (some of them built even little bridges of paper for the broken lines between equal object names to avoid misinterpretations) (cf. [SW99]).

M12.2 Derivation of Judgments from Power Context Families: It is worthwhile to understand the relations in a conceptual graph also as concepts of suitably chosen contexts. This understanding is basic for the derivation of judgments, represented by conceptual graphs, from so-called *power context families* which are composed by contexts $\mathbb{K}_0, \mathbb{K}_1, \mathbb{K}_2, \mathbb{K}_3, \ldots$ where $\mathbb{K}_0$ yields the concepts in the boxes and $\mathbb{K}_1, \mathbb{K}_2, \mathbb{K}_3, \ldots$ yield the concepts of relations of arity $k = 1, 2, 3, \ldots$ in the ovals, respectively (cf. M12.1); clearly, the objects of the relational context $\mathbb{K}_k$ $(k \geq 1)$ are sequences of k objects belonging to the basic context $\mathbb{K}_0$, while the attributes of $\mathbb{K}_k$ have the function to give meaning to those object sequences (cf. [PW99]).

The sketched method can be effectively applied to develop information systems based on power context families representing the relevant knowledge. Such systems have been designed for flight information in Austria and Australia, respectively. The central idea of those information systems is to present to the user, who has inputed his constraints, a conceptual graph representing all flights which might be still relevant. In [PW99], Fig.6, a well-readable output graph is shown to a person who lives in Innsbruck and works in Vienna where he wants to arrive between 7 and 9 a.m. and to depart between 5 and 7 p.m. For more complex requests, the standard diagrams of conceptual graphs might become extremely complicated as shown in [EGSW00], Fig.7, for a customer who lives in Vienna and wants to visit partners in Salzburg, Innsbruck, and Graz at the weekend. But, using background knowledge which can be assumed for the customer, a much better readable diagram of the requested conceptual graph can be offered as shown in [EGSW00], Fig.8. Thus, conceptual graphs should be understood as logical structures which may have many different graphical representations useful for quite different purposes.

3 Supporting Human Thought, Judgment, and Action

As pointed out at the beginning of this paper, the main aim of *Conceptual Knowledge Processing* and its methods is to support rational thought, judgment and action of human beings and to promote the critical discourse. Since Conceptual Knowledge Processing treats knowledge based on actual realities, it relies on the *philosophical logic* as the science of thought in general, its general laws and kinds (cf. [Pe92], p.116). This understanding of philosophical logic has been developed since the 16th century, founded on the doctrines of concept, judgment, and conclusion. The assistance which Conceptual Knowledge Processing obtains from the philosophical logic becomes substantially intensified by the mathematical methods of *Contextual Logic*, which is based on a mathematization of the philosophical doctrines of concept, judgment, and conclusion (cf. [Wi00a]).

Thus, for applying and elaborating the discussed methods of Conceptual Knowledge Processing, it is worth-while not only to work on the level of actual realities in the frame of philosophical logic, but also on the level of potential realities activating mathematical methods. This is, in particular, necessary for the development of new software and theoretical extensions. Nevertheless, the *logical level* should have the primacy over the *mathematical level* because applying methods of Conceptual Knowledge Processing should primarily support human thought, judgment, and action.

Methods of knowledge processing always presuppose, consciously or unconsciously, some understanding of what knowledge is. Different from *ambitious knowledge*, specified for Conceptual Knowledge Processing in Section 1, a quite dominant understanding views knowledge as a collection of facts, rules, and procedures justifiable by objectively founded reasoning. K.-O. Apel criticizes this cognitive-instrumental understanding and advocates for a *pragmatic understanding of knowledge*:

> "In view of this problematic situation [of rational argumentation] it is more obvious not to give up reasoning entirely, but rather to break with the concept of reasoning which is orientated by the pattern of logic-mathematical proofs. In accordance with a new foundation of critical rationalism, Kant's question of transcendental reasoning has to be taken up again as the question about the normative conditions of the possibility of discursive communication and understanding (and therewith discursive criticism too). Reasoning then appears primarily not as deduction of propositions out of propositions within an objectivizable system of propositions in which one has already abstracted from the actual pragmatic dimension of argumentation, but as answering of why-questions of all sorts within the scope of argumentative discourse." (cf. [Ap89], p.19)

In [Wi96], a restructuring of mathematical logic is proposed which locates reasoning within the intersubjective community of communication and argumentation. Only the process of discourse and understanding in the intersubjective community leads to comprehensive states of rationality. Such process does not exclude logic-mathematical proofs, but they can be only part of a broader argumentative discourse (cf. [Wi97b]).

Methods of Conceptual Knowledge Processing can only be successfully applied if *discourses* can be made possible which allow the users and the persons concerned to understand and even to criticize the methods, their performances, and their effects. This does not mean an understanding of all technical details, but the gained competence to judge about the effects which the involved persons and institutions have to expect. A method of conceptual knowledge processing should be transparent in such a manner that persons affected could even successfully fight against the use of that method. An important precondition for critical discourses is that the methods can be communicated in a language which can be understood by the persons concerned; but establishing such languages needs *transdisciplinary efforts* (cf. [Wi02]).

References

[Ap76] K.-O. Apel: Das Apriori der Kommunikationsgemeinschaft und die Grundlagen der Ethik. In: *Transformation der Philosophie*. Band 2: Das Apriori der Kommunikationsgemeinschaft. Suhrkamp Taschenbuch Wissenschaft 165, Frankfurt 1976.

[Ap89] K.-O. Apel: Begründung. In: H. Seiffert, G. Radninzky (Hrsg.): *Handlexikon der Wissenschaftstheorie*. Ehrenwirth, München 1989, 14–19.

[BS97] J. Barwise, J. Seligman: *Information flow: the logic of distributive systems.* Cambridge University Press, Cambridge 1997.

[BH05] P. Becker, J. Hereth Correia: The ToscanaJ Suite for implementing conceptual information systems. In: [GSW05], 324–348.

[Br94] R. B. Brandom: *Making it explicit. Reasoning, representing, and discursive commitment.* Havard University Press, Cambridge 1994.

[Bu00] P. Burmeister: ConImp - Ein Programm zur Formalen Begriffsanalyse. In: [SW00], 25–56.

[CR04] C. Carpineto, G. Romano: *Concept data analysis: theory and applications.* Wiley, London 2004.

[CR05] C. Carpineto, G. Romano: Using concept lattices for text retrieval and mining. In: [GSW05], 161–179.

[EGSW00] P. W. Eklund, B. Groh, G. Stumme, R. Wille: A contextual-logic extension of TOSCANA. In: B. Ganter, G. Mineau (eds.): *Conceptual structures: logical, linguistic and computational issues.* LNAI **1867**. Springer, Heidelberg 2000, 453-467.

[EKSW00] D. Eschenfelder, W. Kollewe, M. Skorsky, R. Wille: Ein Erkundungssystem zum Baurecht: Methoden der Entwicklung eines TOSCANA-Systems. In: [SW00], 254–272.

[FW89] S. Felix, R. Wille: Phonemes of French in Switzerland: an application of Formal Concept Analysis. Unpublished material. TH Darmstadt 1989.

[Ga87] B. Ganter: Algorithmen zur Formalen Begriffsanalyse. In: B. Ganter, R. Wille, K. E. Wolff (Hrsg.): *Beiträge zur Begriffsanalyse.* B.I.-Wissenschaftsverlag, Mannheim 1986, 241–254.

[GSW05] B. Ganter, G. Stumme, R. Wille (eds.): *Formal Concept Analysis: foundations and applications.* State-of-the-Art Survey. LNAI **3626**. Springer, Heidelberg 2005.

[GW89] B. Ganter, R. Wille: Conceptual scaling. In: F. Roberts (ed.): *Applications of combinatorics and graph theory in the biological and social sciences.* Springer-Verlag, New York 1989, 139–167.

[GW99a] B. Ganter, R. Wille: *Formal Concept Analysis: mathematical foundations.* Springer, Heidelberg 1999.

[GW99b] B. Ganter, R. Wille: Contextual Attribute Logic. In: W. Tepfenhart, W. Cyre (eds.): Conceptual structures: standards and practices. LNAI **1640**. Springer, Heidelberg 1999 , 377–388.

[GSJ86] R. Godin, E. Saunders, J. Jecsei: Lattice model of browsable data spaces. *Journal of Information Sciences* **40** (1986), 89–116.

[GD86] J.-L. Guigues, V. Duquenne: Familles minimales d'implications informative resultant d'un tableau de données binaires. Math. Sci. Humaines 95 (1986), 5–18.

[Ha81] J. Habermas: *Theorie des kommunikativen Handelns.* 2 Bände. Suhrkamp, Frankfurt 1981.

[Ks05] T. Kaiser: A TOSCANA-System for the International Regimes Database (IRD). In: H. Breitmeier, O. R. Young, M. Zürn (eds.): *Analyzing international environmental regimes: from case study to database* (to appear)

[Ka88] I. Kant: *Logic*. Dover, New York 1988.

[Kc79] G. L. Karcher: *Konstrastive Untersuchungen von Wortfeldern im Deutschen und Englischen*. Peter Lang, Frankfurt 1979.

[Kr99] M. Karl: *Eine Logik verteilter Systeme und deren Anwendung auf Schaltnetzwerke*. Diplomarbeit. FB Mathematik, TU Darmstadt 1999.

[KW86] U. Kipke, R. Wille: Begriffsverbände als Ablaufschemata zur Gegenstandsbestimmung. In: P. O. Degens, H.-J. Hermes, O. Opitz (Hrsg.): *Die Klassifikation und ihr Umfeld*. Indeks Verlag, Frankfurt 1986, 164–170.

[KW87] U. Kipke, R. Wille: Formale Begriffsanalyse erläutert an einem Wortfeld. *LDV-Forum* **5** (1987), 31–36.

[Ko05] B. Koester: FooCA: Enhacing Google information research by means of Formal Concept Analysis. Preprint. TU Darmstadt 2005.

[KV00] B. Kohler-Koch, F. Vogt: Normen- und regelgeleitete internationale Kooperationen - Formale Begriffsanalyse in der Politikwissenschaft. In: [SW00], 325–340.

[KSVW94] W. Kollewe, M. Skorsky, F. Vogt, R. Wille: TOSCANA - ein Werkzeug zur begrifflichen Analyse und Erkundung von Daten. In: R. Wille, M. Zickwolff (Hrsg.): *Begriffliche Wissensverarbeitung - Grundfragen und Aufgaben*. B.I.-Wissenschaftsverlag, Mannheim 1994, 267–288.

[LS05] L. Lakhal, G. Stumme: Efficient mining of association rules based on Formal Concept Analysis. In: [GSW05], 180–195.

[Lo84] K. Lorenz: Methode. In: J. Mittelstraß (Hrsg.): *Enzyklopädie Philosophie und Wissenschaftstheorie*. Bd.2. B.I.-Wissenschaftsverlag, Mannheim 1984, 876–879.

[Ma00] K. Mackensen: *Simplizität - Genese und Wandel einer musikästhetischen Kategorie des 18. Jahrhunderts*. Bärenreiter, Kassel 2000.

[MW99] K. Mackensen, U. Wille: Qualitative text analysis supported by conceptual data systems. *Quality & Quantity* **33** (1999), 135–156.

[Na96] NaviCon GmbH: Anaconda für Windows. Frankfurt 1996.

[Pe35] Ch. S. Peirce: *Collected papers*. Harvard Univ. Press, Cambridge 1931–35.

[Pe92] Ch. S. Peirce: *Reasoning and the logic of things*. Edited by K. L. Ketner; with an introduction by K. L. Ketner and H. Putnam. Havard University Press, Cambridge 1992.

[PW99] S. Prediger, R. Wille: The lattice of concept graphs of a relationally scaled context. In: W. Tepfenhart, W. Cyre (eds.): *Conceptual structures: standards and practices*. LNAI **1640**. Springer, Heidelberg 1999, 401-414.

[RW00] T. Rock, R. Wille: Ein TOSCANA-Erkundungssystem zur Literatursuche. In: [SW00], 239–253.

[RS84] S. Rückl, G. Schmoll (Hrsg.): *Lexikon der Information und Dokumentation*. VEB Bibliographisches Institut, Leipzig 1984.

[SW99] F. Siebel, R. Wille: Unpublished material. TU Darmstadt 1999.

[So84] J. F. Sowa: *Conceptual structures: information processing in mind and machine*. Adison-Wesley, Reading 1984.

[So92] J. F. Sowa: Conceptual Graphs summary. In: T. E. Nagle, J. A. Nagle, L. L. Gerholz, P. W. Eklund (eds.): *Conceptual structures: current research and practice*. Ellis Horwood, 1992, 3–51.

[SW86] J. Stahl, R. Wille: Preconcepts and set representations of contexts. In: W. Gaul, M. Schader (eds.): *Classification as a tool of research*. North-Holland, Amsterdam 1986, 431–438.

[SW92] S. Strahringer, R. Wille: Towards a structure theory of ordinal data. In: M. Schader (ed.): *Analyzing and modeling data and knowledge*. Spinger, Heidelberg 1992, 129–139.

[SW93] S. Strahringer, R. Wille: Conceptual clustering via convex-ordinal structures. In: O. Opitz, B. Lausen, R. KLar (eds.): *Information and classification. Concepts, methods and applications*. Springer, Heidelberg 1993, 85–98.

[SWW01] S. Strahringer, R. Wille, U. Wille: Mathematical support for empirical theory building. In: H. S. Delugach, G. Stumme (eds.): *Conceptual structures: broadening the base*. LNAI 2120. Springer, Heidelberg 2001, 169–186.

[SC90] A. Strauss. J. Corbin: Basics of qualitative research: grounded theory procedures and techniques. Sage Publ., Newbury Park 1990.

[St97] G. Stumme: *Concept exploration: knowledge acquisition in conceptual knowledge systems*. Dissertation, TU Darmstadt. Shaker Verlag, Aachen 1997.

[SW00] G. Stumme, R. Wille (Hrsg.): *Begriffliche Wissensverarbeitung: Methoden und Anwendungen*. Springer, Heidelberg 2000.

[VW95] F. Vogt, R. Wille: TOSCANA – A graphical tool for analyzing and exploring data. In: R. Tamassia, I. G. Tollis (eds.): *Graph drawing '94*. LNCS **894**. Springer, Heidelberg 1995, 226–233.

[Wi84] R. Wille: Liniendiagramme hierarchischer Begriffssysteme. In: H. H. Bock (Hrsg.): Anwendungen der Klassifikation: Datenanalyse und numerische Klassifikation. Indeks-Verlag, Frankfurt 1984, 32–51; English translation: Line diagrams of hierachical concept systems. *International Classiffication* **11** (1984), 77–86.

[Wi87] R. Wille: Bedeutungen von Begriffsverbänden. In: B. Ganter, R. Wille, K. E. Wolff (Hrsg.): *Beiträge zur Begriffsanalyse*. B.I.-Wissenschaftsverlag, Mannheim 1986, 161–211.

[Wi89a] R. Wille: Lattices in data analysis: how to draw them with a computer. In: I. Rival (Ed.): *Algorithms and order*. Kluwer, Dordrecht 1989, 33–58.

[Wi89b] R. Wille: Knowledge acquisition by methods of Formal Concept Analysis. In: E. Diday (ed.): *Data analysis and learning symbolic and numeric knowledge*. Nova Science Publisher, New York–Budapest 1989, 365–380.

[Wi92] R. Wille: Concept lattices and conceptual knowledge systems. *Computers & mathematics with applications* **23** (1992), 493–515.

[Wi94] R. Wille: Plädoyer für eine philosophische Grundlegung der Begrifflichen Wissensverarbeitung. In: R. Wille, M. Zickwolff (Hrsg.): *Begriffliche Wissensverarbeitung — Grundfragen und Aufgaben*. B.I.-Wissenschaftsverlag, Mannheim 1994, 11–25.

[Wi96] R. Wille: Restructuring mathematical logic: an approach based on Peirce's pragmatism. In: A. Ursini, P. Agliano (eds.): *Logic and algebra*. Marcel Dekker, New York 1996, 267–281.

[Wi97a] R. Wille: Conceptual graphs and formal concept analysis. In: D. Lukose, H. Delugach, M. Keeler, L. Searle, J. F. Sowa (eds.): *Conceptual structures: fulfilling Peirce's dream*. LNAI **1257**. Springer, Heidelberg 1997, 290–303.

[Wi97b] R. Wille: Conceptual landscapes of knowledge: a pragmatic paradigm for knowledge processing. In: G. Mineau, A. Fall (eds.): *Proceedings of the International Symposium on Knowledge Representation, Use, and Storage Efficiency.* Simon Fraser University, Vancouver 1997, 2–13; also in: W. Gaul, H. Locarek-Junge (Eds.): Classification in the Information Age. Springer, Berlin-Heidelberg 1999, 344–356.

[Wi00a] R. Wille: Contextual Logic summary. In: G. Stumme (ed.): *Working with conceptual structures. Contributions to ICCS 2000.* Shaker-Verlag, Aachen 2000, 265–276.

[Wi00b] R. Wille: Begriffliche Wissensverarbeitung: Theorie und Praxis. *Informatik Spektrum* **23** (2000), 357–369; revised version in: B. Schmitz (Hrsg.): *Thema Forschung: Information, Wissen, Kompetenz.* Heft 2/2000, TU Darmstadt, 128–140.

[Wi01] R. Wille: Mensch und Mathematik: Logisches und mathematisches Denken. In: K. Lengnink, S. Prediger, F. Siebel (Hrsg.): *Mathematik und Mensch: Sichtweisen der Allgemeinen Mathematik.* Verlag Allgemeine Wissenschaft, Mühltal 2001, 141–160.

[Wi02] R. Wille: Transdisziplinarität und Allgemeine Wissenschaft. In: H. Krebs, U. Gehrlein, J. Pfeifer, J. C. Schmidt (Hrsg.): *Perspektiven Interdisziplinärer Technikforschung: Konzepte, Analysen, Erfahrungen.* Agenda-Verlag, Münster 2002, 73–84.

[Wi04a] R. Wille: Truncated distributive lattices: conceptual structures of simple-implicational theories. Order **20** (2004), 229–238.

[Wi04b] R. Wille: Dyadic mathematics - abstractions of logical thought. In: K. Denecke, M. Erné, S. L. Wismath (eds.): *Galois connections and applications.* Kluwer, Dordrecht 2004, 453–498.

[Wi05a] R. Wille: Mathematik präsentieren, reflektieren, beurteilen. In: K. Lengnink, F. Siebel (Hrsg.): *Mathematik präsentieren, reflektieren, beurteilen.* Verlag Allgemeine Wissenschaft, Mühltal 2005, 3–19.

[Wi05b] R. Wille: Formal Concept Analysis as mathematical theory of concepts and concept hierarchies. In: [GSW05], 1–33.

[WW96] R. Wille, U. Wille: Coordinatization of ordinal structures. *Order* **13**, (1996), 281–294.

[WW04] R. Wille, U. Wille: Restructuring General Geometry: measurement and visualization of spatial structures. *Contributions to General Algebra* **14**. Johannes Heyn Verlag, Klagenfurt 2004, 189–203.

[Ye00] S. Yevtushenko: System of data analysis “Concept Explorer”. In: *Proceedings of the sevens national conference on Artificial Intelligence KII–2000*, Russia 2000, 127–134. (in Russian)