Off to New Shores
Conceptual Knowledge Discovery and Processing

Gerd Stumme

Institut für Angewandte Informatik und Formale Beschreibungsverfahren AIFB,
Universität Karlsruhe, D–76128 Karlsruhe, Germany
www.aifb.uni-karlsruhe.de/WBS/gst; stumme@aifb.uni-karlsruhe.de

Abstract. In the last years, the main orientation of Formal Concept Analysis (FCA) has turned from mathematics towards computer science. This article provides a review of this new orientation and analyzes why and how FCA and computer science attracted each other. It discusses FCA as a knowledge representation formalism using five knowledge representation principles provided by Davis, Shrobe, and Szolovits [DSS93]. It then studies how and why mathematics-based researchers got attracted by computer science. We will argue for continuing this trend by integrating the two research areas FCA and Ontology Engineering.
The second part of the article discusses three lines of research which witness the new orientation of Formal Concept Analysis: FCA as a conceptual clustering technique and its application for supporting the merging of ontologies; the efficient computation of association rules and the structuring of the results; and the visualization and management of conceptual hierarchies and ontologies including its application in an email management system.

Content

1 Introduction

Formal Concept Analysis (FCA) has observed a major change of orientation in the last years. Having been introduced as a mathematization of the concept of ‘concept’ in the early 1980ies, its main orientation has turned from mathematics towards computer science during the last ten years: ten years ago, virtually all FCA papers were given at mathematics conferences, while nowadays they are given almost exclusively at conferences related to computer science. FCA is now considered as the mathematical backbone of *Conceptual Knowledge Processing (CKP)*, a theory located in computer science, having as task to provide methods and tools for human–oriented, concept–based knowledge processing.

In this article, the change of orientation will be reviewed from a subjective point of view. During his stay at the Department of Mathematics at Darmstadt University of Technology and at computer science groups at Blaise Pascal University, Clermont-Ferrand, and the University of Karlsruhe, the author has observed and also actively shaped this new orientation. It will be analyzed why FCA became attractive as a knowledge representation method for computer science, and why computer science became attractive for researchers working on FCA.

After a recall of some FCA basics in the next section, we start with the analysis of why FCA is a suitable knowledge representation formalism, based on the article "What is a knowledge representation?" by R. Davis, H. Shrobe, and P. Szolovits [DSS93]. According to the authors, a knowledge representation is (i) a medium of human expression, (ii) a set of ontological commitments, (iii) a surrogate, (iv) a fragmentary theory of intelligent reasoning, and (v) a medium for pragmatically efficient computation. We will show that principles (i) to (iii) have been discussed intensively for FCA and can even be considered as its driving forces. Although not completely neglected, principles (iv) and (v) were not in the main focus of FCA in early times. They received increased attention when FCA turned towards computer science, and led to the research area of Conceptual Knowledge Processing.

Having analyzed the attractiveness of FCA as a knowledge representation method for computer science, we will discuss why computer science became attractive for researchers working on FCA. One important reason will turn out to be that computer science showed to be much more open for a discussion of its justifications, expectations, and possible consequences — a discussion claimed necessary by H. von Hentig, whose program of restructuring sciences [Hn74] was the main impulse to develop FCA in order to restructure mathematical lattice theory. Another reason for the change of orientation will turn out to be that the research questions attacked by mathematics-based FCA researchers started to differ from those of main stream mathematics and found a new home in computer science.

The new home is *Conceptual Knowledge Processing*. Its aim is to provide methods and tools for acquiring, reasoning with, and representing knowledge, and for making it available to human communication. As concepts are the basic units of human thought, CKP is based on theories for modeling concepts. Primarily, this has been Formal Concept Analysis, but nowadays Conceptual Graphs, are also considered, as well as links to Description Logics; and we will argue for a future integration of ontologies. Currently, two main research trends can be distinguished in CKP: Contextual Logic and Conceptual Knowledge

Discovery. *Contextual Logic* aims at restructuring mathematical logic, following Hentig's restructuring program. *Conceptual Knowledge Discovery* pursues a human-centered approach to knowledge discovery, based on concept-oriented theories. We will discuss these two research trends, with a focus on the latter.

We conclude with a discussion of some recent work witnessing the new orientation of FCA as grouped together in the author's habilitation thesis [St02a]. The aim is to show how this change of orientation inspired research within the whole bandwidth from theory to applications. We discuss three lines of research:

- the use of FCA as a conceptual clustering technique and its application for supporting the merging of ontologies,
- the efficient computation of association rules, and the structuring and reduction of the results,
- and the visualization and the management of conceptual hierarchies/ontologies, and its application in an email management system.

The article is an extended version of an invited talk given at the 10th International Conference on Conceptual Structures [St02b].

1.1 Organization of the Article

In the next section, the most basic definitions of FCA are recalled, in order to give the reader a taste of this theory. Section 3 provides a discussion about knowledge representation with FCA according to the principles given in [DSS93]. In Section 4 we review the change of orientation of FCA towards computer science. Its extension to Conceptual Knowledge Processing and Discovery is the topic of Section 5.

Sections 6 to 8 present selected applications: Section 6 presents FCA as a conceptual clustering technique and its application for supporting the merging of ontologies; Section 7 shows the use of FCA for the efficient computation of association rules and the structuring and reduction of the results; and Section 8 discusses the visualization and management of conceptual hierarchies including its application in an email management system. Section 9 concludes the article.

2 Formal Concept Analysis: a Mathematization of Concepts

This section is meant to be an 'appetizer' — to give the reader a taste of Formal Concept Analysis. It provides a brief illustration of the core notions. Readers familiar with FCA may skip this section.

Formal Concept Analysis (FCA) was introduced as a mathematical theory modeling the concept of 'concepts' in terms of lattice theory. To allow a mathematical description of extensions and intensions, FCA starts with a *(formal) context*.

Definition 1. *A* (formal) context *is a triple* $\mathbb{K} := (G, M, I)$*, where* G *is a set whose elements are called* objects*,* M *is a set whose elements are called* attributes*, and* I *is a binary relation between* G *and* M *(i. e.* $I \subseteq G \times M$*).* $(g, m) \in I$ *is read "the object* g has *the attribute* m*".*

Figure 1 shows a formal context where the object set G comprises all airlines of the Star Alliance group and the attribute set M lists their destinations. The binary relation I is given by the cross table and describes which destinations are served by which Star Alliance member.

Definition 2. *For* $A \subseteq G$*, let*

$$A' := \{m \in M \mid \forall g \in A \colon (g, m) \in I\}$$

and, for $B \subseteq M$*, let*

$$B' := \{g \in G \mid \forall m \in B \colon (g, m) \in I\} \ .$$

A (formal) concept *of a formal context* (G, M, I) *is a pair* (A, B) *with* $A \subseteq G$*,* $B \subseteq M$*,* $A' = B$ *and* $B' = A$*. The sets* A *and* B *are called the* extent *and the* intent *of the formal concept* (A, B)*, respectively. The subconcept–superconcept relation is formalized by*

$$(A_1, B_1) \leq (A_2, B_2) :\Longleftrightarrow A_1 \subseteq A_2 \quad (\Longleftrightarrow B_1 \supseteq B_2) \ .$$

The set of all formal concepts of a context $\mathbb{K}$ *together with the order relation* $\leq$ *is always a complete lattice,*[1] *called the* concept lattice *of* $\mathbb{K}$ *and denoted by* $\underline{\mathfrak{B}}(\mathbb{K})$*.*

Figure 2 shows the concept lattice of the context in Figure 1 by a *line diagram*. Line diagrams follow the conventions for the visualization of hierarchical concept systems as established in the German standard [DIN2331]. In a line diagram, each node represents a formal concept. A concept $\mathfrak{c}_1$ is a subconcept of a concept $\mathfrak{c}_2$ if and only if there is a path of descending edges from the node representing $\mathfrak{c}_2$ to the node representing $\mathfrak{c}_1$. The name of an object g is always attached to the node representing the smallest concept with g in its extent; dually, the name of an attribute m is always attached to the node representing the largest concept with m in its intent. We can read the context relation from the diagram because an object g has an attribute m if and only if the concept labeled by g is a subconcept of the one labeled by m. The extent of a concept consists of all objects whose labels are attached to subconcepts, and, dually, the intent consists of all attributes attached to superconcepts. For example, the concept labeled by 'Middle East' has {Singapore Airlines, The Austrian Airlines Group, Lufthansa, Air Canada} as extent, and {Middle East, Canada, United States, Europe, Asia Pacific} as intent.

In the top of the diagram, we find the destinations which are served by most of the members: Europe, Asia Pacific, and the United States. For instance, beside

[1] I. e., for each subset of concepts, there is always a unique greatest common subconcept and a unique least common superconcept.

	Latin America	Europe	Canada	Asia Pacific	Middle East	Africa	Mexico	Caribbean	United States
Air Canada	×	×	×	×	×		×	×	×
Air New Zealand		×		×					×
All Nippon Airways		×		×					×
Ansett Australia				×					
The Austrian Airlines Group		×	×	×	×	×			×
British Midland		×							
Lufthansa	×	×	×	×	×	×	×		×
Mexicana	×		×				×	×	×
Scandinavian Airlines	×	×		×		×			×
Singapore Airlines		×	×	×	×	×			×
Thai Airways International	×	×		×				×	×
United Airlines	×	×	×	×			×	×	×
VARIG	×	×		×		×	×		×

Fig. 1. A formal context about the destinations of the Star Alliance members

British Midland and Ansett Australia, all airlines are serving the United States. Those two airlines are located at the top of the diagram, as they serve the fewest destinations — they operate only in Europe and Asia Pacific, respectively.

The further we go down in the concept lattice, the more globally operating are the airlines. The most destinations are served by the airlines at the bottom of the diagram: Lufthansa (serving all destinations beside the Caribbean) and Air Canada (serving all destinations beside Africa). Also, the further we go down in the lattice, the fewer served are the destinations. For instance, Africa, the Middle East, and the Caribbean are served by relatively few Star Alliance members.

Dependencies between the attributes can be described by implications. For $X, Y \subseteq M$, we say that the *implication* $X \rightarrow Y$ *holds* in the context, if each object having all attributes in X also has all attributes in Y. For instance, the implication {Europe, United States} $\rightarrow$ {Asia Pacific} holds in the Star Alliance context. It can be read directly in the line diagram: the largest concept having both 'Europe' and 'United States' in its intent (i. e., the concept labeled by 'All Nippon Airways' and 'Air New Zealand') also has 'Asia Pacific' in its intent. Similarly, one can detect that the destinations 'Africa' and 'Canada' together imply the destination 'Middle East' (and also 'Europe', 'Asia Pacific', and 'United States').

Formal Concept Analysis is also able to deal with many-valued contexts, i. e., contexts may not only have binary attributes, but attribute-value pairs. A *many-valued context* is a tuple $\mathbb{K} := (G, M, (W_m)_{m \in M}, I)$ where G is a set of objects, M a set of attributes, W_m the set of possible values for the attribute $m \in M$, and the relation $I \subseteq G \times \{(m, w) \mid m \in M, w \in W_m\}$, with the constraint

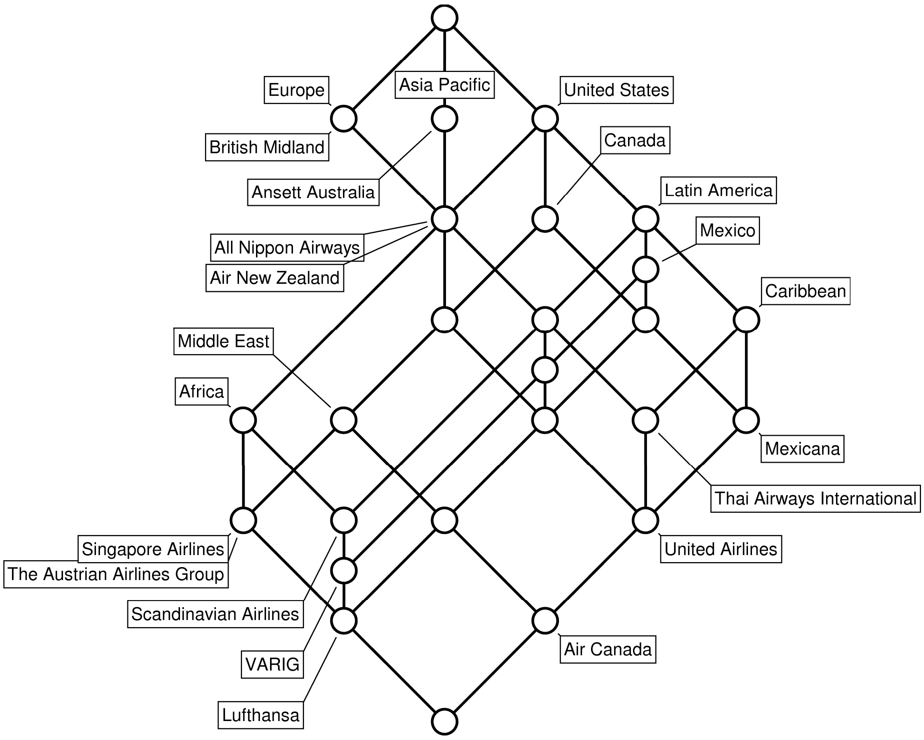

Fig. 2. The concept lattice of the context in Figure 1

$(g, m, w_1) \in I, (g, m, w_2) \in I \Rightarrow w_1 = w_2$ imposed, indicates if an object $g \in G$ has value $w \in W_m$ for attribute $m \in M$.

Conceptual scales reflect each a different view on the data. They are used (i) to transform many-valued contexts (i. e., contexts consisting of object–attribute–value triples) into one-valued contexts and (ii) to split large contexts into 'vertical slices' having concept lattices of manageable size (see also 'horizontal reduction' in Section 6.1).

A *conceptual scale* for a subset $B \subseteq M$ of attributes is a (one-valued) formal context $\mathbb{S}_B := (G_B, M_B, I_B)$ with $G_B \subseteq \times_{m \in B} W_m$. The idea is to replace the attribute values in W_m which are often too specific by more general attributes which are provided in M_B:

Let $\mathfrak{S}$ be the set of conceptual scales for the many-valued context $\mathbb{K} := (G, M, (W_m)_{m \in M}, I)$. For any subset $\mathcal{S} \subseteq \mathfrak{S}$ of scales, we can now translate the many-valued context into a one-valued one: The *derived context* $\mathbb{K}_\mathcal{S}$ is defined by $\mathbb{K}_\mathcal{S} := \big(G, \bigcup_{\mathbb{S}_B \in \mathcal{S}} M_B, I_\mathcal{S}\big)$ with $(g, n) \in I_\mathcal{S}$ if there exists a scale $\mathbb{S}_B \in \mathcal{S}$ with $m \in M_B$ and $w \in W_m$ where $(g, m, w) \in I$ and $(g, n) \in I_B$. We will give an example for a special case below.

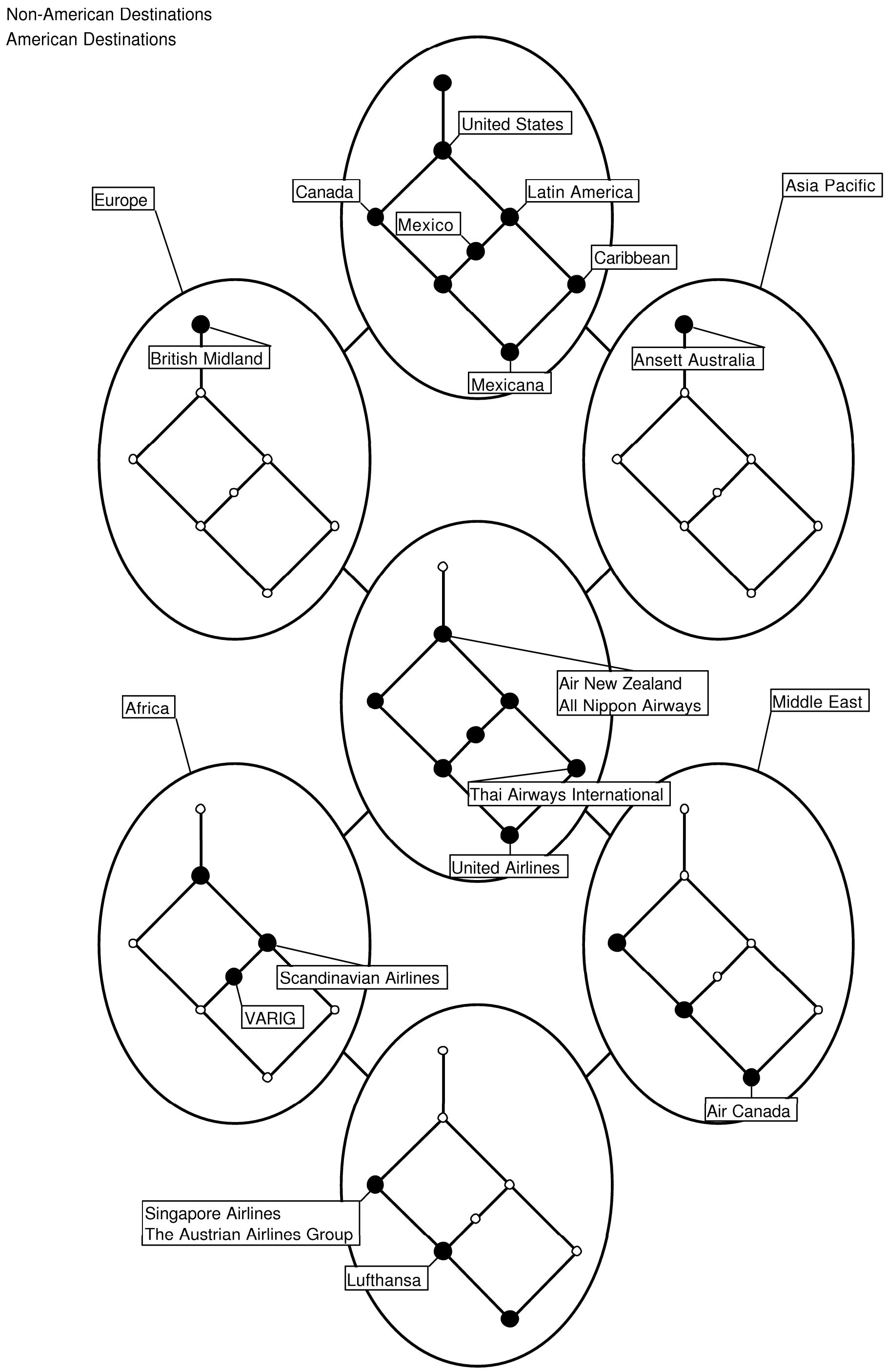

Fig. 3. A nested line diagram of the concept lattice in Figure 2

If $\mathcal{S}$ consists of two (or more scales), the concept lattice of the derived context can be visualized in a *nested line diagram* (see Figure 3). In nested line diagrams, the nodes of the concept lattice of the first scale are enlarged, so that the concept lattice of the second scale can be drawn inside. The second lattice is then used to further differentiate each of the extents of the concepts of the first lattice. Conceptual scaling and nested line diagrams are for instance implemented in the management system TOSCANA for conceptual information systems [KSVW94,VW95]. A *conceptual information system* consists of a many-valued context and a set of conceptual scales.

Conceptual scaling will not be discussed in general in this article; we restrict ourselves to a sub-case: (One-valued) formal contexts are specific instances of many-valued contexts, one just has $W = \{\text{yes}, \text{no}\}$. Conceptual scaling can also be applied to this special situation, in order to obtain nested line diagrams.

Figure 3 shows a nested line diagram for the Star Alliance context. It is obtained by the definitions above letting $B_1 := \{\,$Europe, Asia Pacific, Africa, Middle East$\,\}$, $B_2 := \{\,$United States, Canada, Latin America, Mexico, Caribbean$\,\}$, $\mathbb{S}_i := (\mathfrak{P}(B_i), B_i, \ni)$, for $i \in \{1, 2\}$, and $\mathcal{S} := \mathfrak{S} := \{\mathbb{S}_1, \mathbb{S}_2\}$. The diagram shows the direct product of the concept lattices of the scales $\mathbb{S}_1$ (as large diagram) and $\mathbb{S}_2$ (as the inner diagrams). Its order relation can be read by replacing each of the lines of the large diagram by eight parallel lines linking corresponding nodes in the inner diagrams. The concept lattice given in Figure 2 is embedded (as a join–semilattice) in this diagram, it consists of the bold nodes. The concept mentioned above (labeled by 'Middle East') is for instance represented by the left-most bold node in the lower right part.

The bold concepts are referred to as 'realized concepts', as their intents correspond to intents of a concept of the realized scale. The non-realized concepts are not only displayed to indicate the structure of the inner scale, but also because they indicate implications: Each non-realized concept indicates that the attributes in its intent imply the attributes contained in the largest realized concept below. For instance, the first implication discussed above is indicated by the non-realized concept having as intent 'Europe' and 'United States', it is represented by the empty node below the concept labeled by 'British Midland'. The largest realized sub-concept below is the one labeled by 'All Nippon Airways' and 'Air New Zealand' — which additionally has 'Asia Pacific' in its intent. Hence $\{$ Europe, United States $\} \to \{$ Asia Pacific $\}$ holds. The second implication from above is indicated by the non-realized concept left of the concept labeled by 'Scandinavian Airlines', and the largest realized concept below, which is the one labeled by 'Singapore Airlines' and 'The Austrian Airlines Group'.

This section gave a short introduction to the core notions of FCA. Of course, there exist more complex (data) structures capable to represent further relevant aspects of conceptual knowledge. We refer to some of them in the subsequent discussion.

3 Knowledge Representation with Formal Concept Analysis

The convergence of FCA with computer science demands for a discussion about their relationships. In [Zi92], [WZ94], [SWW98], [MSW99], [HSWW00], and [Wi01b], several aspects of this relationship have been studied. In this article we take up the discussion. In [DSS93], R. Davis, H. Shrobe, and P. Szolovits studied the question "What is a knowledge representation?" They provided five principles a knowledge representation should follow. Together with a list of representation levels (of semantic networks) provided by R. Brachman in [Bra79], we will use these principles to "characterize and make explicit the 'spirit' of [Formal Concept Analysis], the important set of ideas and inspirations that lie behind [...] the concrete machinery used to implement the representation." [DSS93]. Davis *et al*'s principles are as follows:[2]

1. A knowledge representation "is a *medium of human expression*, i. e., a language in which we say things about the world."
2. "It is a *set of ontological commitments*, i. e., an answer to the following question: In what terms should I think about the world?"
3. It "is most fundamentally a *surrogate*, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i. e., by reasoning about the world rather than taking action in it."
4. "It is a *fragmentary theory of intelligent reasoning*, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation *sanctions*; and (iii) the set of inferences it *recommends*."
5. "It is a *medium for pragmatically efficient computation*, i. e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences."

The authors claim that these principles offer a framework for making explicit the 'spirit' of a representation, and the way it emphasizes on one or more of them characterizes the fundamental 'mindset' of the representation. Each knowledge representation formalism is in some way a trade-off between these principles. We will use these five criteria for discussing the role of FCA as knowledge representation method.

It will turn out that the first three principles (especially the first one) have been the driving forces for the development of FCA, while interest on the last two principles — although not completely absent at the beginning (see for instance knowledge acquisition with attribute exploration, implicational theories, and efficient computation of concept lattices [Ga87]) — increased during the change of orientation of FCA towards computer science.

[2] Davis *et al* discuss these principles in the order 3–2–4–5–1. Here we reorder them to follow more closely the historical development of FCA.

3.1 FCA as a medium of human expression

"Knowledge representations are [...] the medium of expression and communication in which we tell the machine (and perhaps one another) about the world. [...] Knowledge representation is thus a medium of expression and communication for the use by *us*" [DSS93]. In other words: "A representation is the language in which we communicate, hence we must be able to speak it without heroic effort".

This observation has always been predominant for the development of theory for and applications of FCA, as the strong emphasis on its philosophical roots shows. When introducing FCA in [Wi82], R. Wille's purpose was to restructure lattice theory: "*Restructuring lattice theory* is understood as an attempt to unfold lattice-theoretical concepts, results, and methods in a continuous relationship with their surroundings [...]. One basic aim is to promote better communication between lattice theorists and potential users of lattice theory" [Wi82, pp. 447]. The program of restructuring lattice theory followed a programatic discussion about the role of sciences in our society by H. von Hentig [Hn74]. Hentig requests that the sciences "uncover their non-intended aims, declare their intended aims, select and adjust their means according to those aims, discuss openly and understandably their justifications, expectations, and possible consequences, and therefore disseminate their means of research and results in common language" [Hn74, pp. 136 f; translated by the author]. As application, Wille referred to the roots of the lattice idea, namely hierarchies of concepts, which played an important role in attempts to formalize logic [Sc90]. Wille discusses in his visionary article "how parts of arithmetic, structure and representation theory of lattices may be developed out of problems and questions which occur within the analysis of contexts and their hierarchies of concepts" [Wi82, pp. 448].

A second philosophical foundation of FCA is the pragmatic philosophy of Ch. S. Peirce [Pe31], and the Theory of Communicative Action of J. Habermas [Ha81] (cf. [Wi94,Wi99]). Peirce considers knowledge as always incomplete, formed and continuously assured by human discourse. J. Habermas took up these ideas in his Theory of Communicative Action where he emphasizes on the importance of the inter-subjective community of communication. He observes that humans operate in argumentative dispute on the normative basis of practical-ethical rules. Even in scientific statements (i. e., in assertions), one tries to convince the listener and expects agreement or counter-arguments. Hence even in these apparently objective domains the ethical norms of equality and acceptance are thus present (cf. [Ho95, p. 338]). Following this line of argumentation, the task for theories formalizing aspects of knowledge is thus to provide means for rational communication.

The observation that this understanding conflicts with the widely accepted view of mathematics as a means for mechanistic problem solving was certainly one of the main reasons for the change of orientation of FCA towards computer science, where human(–computer) interaction is considered as a research topic on its own (although large parts of computer science also follow a rather mechanistic view).

In their discussion, Davis, Shrobe, and Szolovits emphasize two questions: “How well does [a knowledge representation] function as a medium of communication? [...] What kinds of things are easily said in the language and what kinds of things are so difficult as to be pragmatically impossible?” In FCA, the representation one starts with is quite simple: formal contexts are sets of object–attribute pairs (or sets of object–attribute–value triples in the case of many-valued contexts). This representation allows for an easy representation of discrete structures, while continuous structures (e. g., time, space) are representable with limitations only. The advantage of this relatively simple representation as starting point is that it is understandable even by untrained users, while the resulting concept lattices allow to gain deep and surprising insights into the knowledge implicitly contained in the data. Experiences in different applications (e. g., the use of FCA in the treatment of patients suffering from bulimia who have no higher mathematical education [SpW89]) shows the acceptance of FCA by untrained users, based on the visualization of the data by concept lattices. This is why FCA has been shown to be useful for Knowledge Discovery, as we will discuss in Section 5.2.

3.2 The ontological commitment of FCA

“In selecting any representation, we are [...] making a set of decisions about how and what to see in the world. [...] We (and our reasoning machines) need guidance in deciding what in the world to attend to and what to ignore” [DSS93]. Formal Concept Analysis formalizes the concepts concept, concept extension, concept intension, and conceptual hierarchy. We discuss this ontological commitment of FCA along three lines: a definition of concept given in a philosophical lexicon, the international standard ISO 704, and a list of representation levels for semantic networks provided by R. Brachman.

> **Concept.** A concept is the most basic unit of thought, in contrast to judgment and conclusion, which are forms of thought composed of concepts. While a judgment makes an assertion about an issue, a concept is a notional, i. e., abstract–mental, representation of its ‘whatness’; it captures an object based on ‘what’ it is, without already making an assertion about it. [...] For each concept one distinguishes its *intension* and *extension.* The intension of a concept comprises all attributes thought with it, the extension comprises all objects for which the concept can be predicated. In general, the richer the intension of a concept is, the lesser is its extension, and vice versa. [Bru76, p. 39f; translated by the author]

This lexicon entry reflects a predominant understanding of concepts as being the most basic units of thought, based on which more complex entities of thought — i. e., judgments and conclusions — can be built. This understanding has grown during centuries from Greek philosophy to late Scholastic and has been stated in modern terms in the 17th century in the Logic of Port Royal [AN85]. It is nowadays established in the standard [ISO704] (which is the international

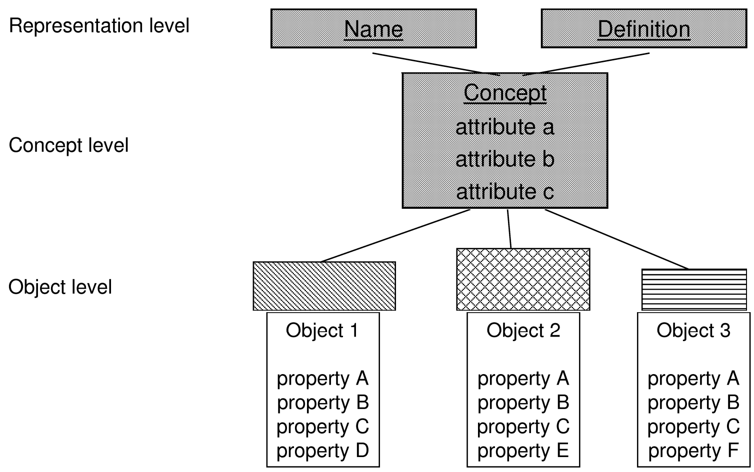

Fig. 4. Object level, concept level, and representation level according to ISO 704

version of the German standard [DIN2330]). The definition of formal concepts as given in Section 2 follows closely this understanding. It explicitly formalizes extension and intension of a concept, their mutual relationships, and the fact that increasing intent implies decreasing extent and vice versa. The formalization of concepts by FCA follows thus a long philosophical tradition.

The standard ISO 704 distinguishes three levels: object level, concept level, and representation level (see Figure 4). There is no immediate relationship between objects and names. This relationship is rather provided by concepts. On the concept level, the objects under discussion constitute the extension of the concept, while their shared properties constitute the intension of the concept. On the representation level, a concept is specified by a definition and is referred to by a name.[3]

While other knowledge representation formalisms like Description Logics or Conceptual Graphs mainly focus on the representation level, the focus of FCA is on the concept level. In fact, the definition of formal concepts follows closely the description of that level in [ISO704]: formal concepts consist of extension and intension (only), while concept names and definitions are not within the (core) notions of FCA. Thus FCA should not be considered as competing with the other mechanisms, but rather as a complement. There is recent work following this view, for instance in combining FCA with Description Logics (e. g., [Ba95], [St96b], [Pr97], [PS99]) or with Conceptual Graphs (e. g., [Wi97], [PW99], see also [MSW99]) leading to the development of Contextual Logic (see Section 5.1).

[3] After a discussion of the three levels, ISO 704 provides an overview over naming and definition principles, and provides quality criteria for them.

The third line for discussing the ontological commitment of FCA is along the list of five representation levels of semantic networks provided by R. Brachman in [Bra79]:[4]

- *Implementational Level*: The primitives are nodes and links where links are merely pointers and nodes are simply destinations for links. On this level there are only data structures out of which to build logical forms.
- *Logical Level*: The primitives are logical predicates, operators, and propositions together with a structured index over those primitives. On this level logical adequacy is responsible for meaningfully factoring knowledge.
- *Epistemological Level*: The primitives are conceptual units, conceptual subpieces, inheritance and structuring relations. On this level conceptual units are determined in their inherent structure and their interrelationships.
- *Conceptual Level*: The primitives are word-senses and case relations, object- and action-types. On this level small sets of language-independent conceptual elements and relationships are fixed from which all expressible concepts can be constructed.
- *Linguistic Level*: The primitives are arbitrary concepts, words, and expressions. On this level the primitives are language-dependent, and are expected to change in meaning as the network grows.

Brachman argues that "any network notation could be analyzed in terms of any of the levels, since they do not have any absolute, independent existence. [...] However, each network scheme does propose an *explicit* set of primitives as part of the language [...]." This set of primitives can be used to classify a network according to the levels.

Here we focus on concept lattices, the core structures of Formal Concept Analysis, which are located on the epistemological level: formal concepts are considered as "formal objects, with predetermined internal organization that is more sophisticated than sets of cases" [Bra79]. Formal concepts bring together extensional and intensional views on 'concepts', and represent explicitly inheritance by referring to the set semantics of the intents (or extents) of the formal concepts. On the formal side, these are made explicit by the following four relations: an object has an attribute ($(g, m) \in I$ in terms of Section 2), an object belongs to a concept ($g \in A$, for a concept (A, B)), an attribute abstracts from a concept ($m \in B$), and a concept is a subconcept of another concept ($(A_1, B_1) \leq (A_2, B_2)$) [LuW91].

Concept lattice constructions also belong to the epistemological level. As Formal Concept Analysis is founded on lattice theory, lattice constructions and lattice decompositions can be activated for establishing more complex concept hierarchies out of simpler ones, and, vice versa, for reducing complex concept hierarchies to simpler ones. Constructions like direct products and tensor products of concept lattices and decompositions like subdirect and atlas decompositions

[4] The purpose of Brachman's paper was to show that those days' semantic networks covered all but the epistemological level; and that his KL-ONE formalism filled the gap.

have been successfully applied in data analysis. It is interesting to note that most concept lattice constructions and decompositions have as counterpart a context construction (which is situated on Brachman's logical level, see [GW99a]).

A more detailed discussion of FCA with respect to Brachman's representation levels is provided in [MSW99] and — especially focussing on Conceptual Knowledge Discovery — in [HSWW00].

3.3 Formal contexts and concepts as surrogates

"Reasoning is a process that goes on internally [of a person or program], while most things it wishes to reason about exist only externally. [...] This unavoidable dichotomy is a fundamental rationale and role for a representation: it functions as a surrogate inside the reasoner" [DSS93]. The authors emphasize that (human or machine) reasoning cannot deal directly with objects in the world, but only with an internal substitute: the knowledge representation.

The basic surrogates in FCA are formal contexts and concept lattices. The notion of *formal contexts* follows the understanding that one can analyze and argue only in restricted contexts, which are always subject to pre-knowledge and social conventions [Wi97]. In applications, the transition from reality to the formal model (and back) is made explicit by the use of formal contexts; such that this interface between reality and model is always open to argumentation. Also *formal concepts*, being surrogates, only consider selected aspects of concepts, excluding for instance fuzzyness, prototypical concepts, modification over time, and so forth. In order to overcome some of the restrictions, there have been developed extensions of the formalism, for instance allowing for fuzzy concepts [Po97] or more expressive intensional descriptions of concepts [Pr97,PS99].

3.4 FCA as fragmentary theory of intelligent reasoning

"The initial conception of a representation is typically motivated by some insight indicating how people reason intelligently, or by some belief about what it means to reason intelligently at all" [DSS93]. The authors consider five fields which have provided notions of what constitutes intelligent reasoning: mathematical logic (e. g., Prolog), psychology (e. g., frames), biology (e. g., neural networks), statistics (e. g., bayesian networks), and economics (e. g., rational agents).

As other knowledge representation formalisms, FCA is opposed to the logistic belief that reasoning intelligently necessarily means reasoning in the fashion defined by first-order logic. The roots of FCA are best described in a philosophical view (which is close to what Davis *et al* describe as "psychological view"). It emphasizes on inter-subjective communication and argumentation, as discussed in Section 3.1. Thus — in contrast to other formalisms — FCA as such (i. e., without its extension to CKP, especially to Contextual Logic) refers the reasoning to the human user who is able to involve common sense, social conventions, views, and purposes. One of the foremost aims of FCA has always been to *support* human thinking, communication, and argumentation rather than *mechanizing* it. In [Wi87,Wi99,Wi01a], Wille discusses the diversity in which intelligent reasoning supported by FCA takes place through sets of real-world applications. FCA in its basic form focuses on reasoning with concepts; its extension to Contextual Logic also provides a theory for reasoning about and with judgments and conclusions, including thus the triad concept–judgment–conclusion of classical philosophical logic (see Section 5.1). Reasoning with concepts comprises for instance implicational theories [Ga87,Wd95,STBPL01a], clauses [GW99b], and hypothesis generation [GK00].

3.5 Efficient computation within FCA

Davis, Shrobe, and Szolovits stress the importance of having a description of a useful way to organize information which allows for suggesting reasoning mechanisms and for facilitating their execution. Even though automatic reasoning is less in the heart of FCA as it is in most other knowledge representation formalisms, the question how to organize information is also important for supporting human reasoning.

In FCA, information is organized in lattices. Lattices provide a clear structure for knowledge representation, which most fundamentally comprises a partial order. Unlike other partial orders (e. g., trees), they allow for multiple inheritance, which often supports a more structured representation and facilitates retrieval of the stored information. Additionally, knowledge representation in lattices is equivalent to apparently unrelated representations such as implications and closure operators. This allows to transfer knowledge into multiple formats each of which is best fit to the actual task. Last but not least, (concept) lattices are equipped with an algebraic structure (stemming from the existence of unique greatest common sub- and least common super-concepts, similar to greatest common divisors and least common multiples for natural numbers) which allows for computation within the lattice structure. As mentioned in Section 3.2, most concept lattice constructions and decompositions have as counterpart a context construction. As formal contexts are only 'logarithmic in size' compared to the concept lattice, they can be seen as a medium of efficient computation.

One can thus exploit the wealth of results of lattice theory for efficient computation. For instance, properties of closure systems are used for computing the concept lattice (e. g., [Ga87], [STBPL02], see also Section 6.1) and valid implications (e. g., [Ga87]); and lattice constructions are used for the efficient visualization by nested line diagrams (e. g., [Wi84], [St96a], see also Section 8.1). Results from lattice theory have also been exploited for data mining tasks, for instance for conceptual clustering (e. g., [StrW93,MG95] and [STBPL02], see also Section 6.1), and for association rule mining (e. g., [STBPL01a], see also Section 7.2). There is still a huge open scientific potential in bringing together structural–mathematical aspects (here especially from FCA) and procedural–computational aspects from computer science.

Having discussed the attractiveness of FCA as a knowledge representation method for computer science, we will study in the next section why and how mathematics-based FCA researchers got attracted by computer science.

4 Off to New Shores

As concepts are the most basic units of thought, it is not surprising that they became important building blocks in Artificial Intelligence (AI) research. Their appearance is prevailing in Knowledge Representation (e. g., in semantic networks, conceptual graphs, description logics), but they also appear for instance in Machine Learning (e. g., in conceptual clustering, concept learning). All these approaches focus on other aspects of concepts, leading to different formalizations.

Formal Concept Analysis arose independently of the formalisms mentioned above. Integrating several ideas from quite different domains (e. g., [Bi40,BM70] [Hn74,DIN2330]), FCA was introduced in 1979 by R. Wille as a *mathematical* theory, in order to "restructure lattice theory", following Hentig's restructuring program (see Section 3.1). A consequence of the aim of restructuring lattice theory was that research in the early time of FCA (1980ies and early 1990ies) mainly fell into three categories: *i*) lattice theory (e. g., lattice constructions and decompositions [Wi83]), *ii*) qualitative data analysis (e. g., a generalized measurement theory [GSW86]), and *iii*) applications (e. g., the analysis of surveys [Ko89]). Of course, algorithms for computing concept lattices also were an important topic (see for instance [Ga87]).

Until the beginning of the 1990ies, the development in AI and in FCA went on almost independently. By then, the mutual perception increased. For instance, FCA researchers got in contact with the knowledge acquisition community, and AI researchers integrated FCA in their approaches (e. g., [CR93]). As discussed in the previous section, FCA became attractive as an AI knowledge representation, and (as we will see below), mathematicians working on FCA got interested in AI research topics. This convergence led to the aim to establish Conceptual Knowledge Processing as an extension of FCA (see next section). In 1993, the ErnstSchröderCenter for Conceptual Knowledge Processing[5] was founded in Darmstadt to support and accompany this development. Just a year later, NaviCon GmbH[6] was founded, a spin-off of Darmstadt University of Technology offering consulting based on FCA methods and tools.

The convergence of FCA with computer science research increased significantly by the series of International Conferences on Conceptual Structures (ICCS), where FCA became a topic in 1995 [LeW95,St95]. This conference series especially stimulated the development of Contextual Logic [Wi96] (see Section 5.1). From 1998 on, the use of FCA for Knowledge Discovery was discussed [SWW98], and FCA was applied for improving the efficiency of data mining algorithms (see Section 7). Today, FCA is not only considered within AI, but also in other computer science domains, as for instance in software engineering

[5] www.mathematik.tu-darmstadt.de/ags/esz/

[6] www.navicon.de

(e. g., [Sn96]) or database theory (e. g., [SS98]). FCA papers are nowadays almost exclusively presented at computer science conferences and in computer science journals. The foundation of the Research Center for Conceptual Knowledge Processing (FZBW)[7] at Darmstadt University of Technology in November 2000 also witnesses the continuous interest in this research topic.

One reason for the change of orientation of FCA (and CKP) towards computer science is certainly that, in the eyes of the mathematical community, lattice theory is an almost closed research area, where almost all important problems have been solved. Further open problems, for instance the development of good lattice drawing algorithms, are not considered as genuine mathematical problems by the majority of the mathematicians.

A more important reason for the change of orientation is the fact, that computer science is — perhaps because it is still a young discipline — in general much more open-minded to discussions such as Hentig's restructuring program than mathematics is. The relationship and the interaction between user and computer is a research domain in computer science for its own sake, and, more important still, expectations and possible consequences of computer science are discussed in public.

What are future directions of Formal Concept Analysis? We conclude this section by relating Conceptual Knowledge Processing with the growing research area of *Ontology Engineering* (see for instance [Mä02]). We believe that nowadays FCA and (parts of) AI are closer together as they sometimes seem to be. This holds especially for the consideration of the importance of the principle of knowledge representation as a medium of human expression. Partly the remaining difference is due only to the different language they (still) speak. In fact, the importance of this principle has increasingly been discussed in the AI community in the past few years.

Interestingly, Ontology Engineering (independently) follows a trend which also served as basis for FCA. The point is that, according to J. Habermas, *ontology*, stemming from the tradition of Greek metaphysics, is constrained to a specific relationship to the world, namely the cognitive relationship to the existing world. It does not consider the subjective nor the social world. A concept corresponding to 'ontology', which includes the relationship to the subjective and social world, as well as to the existing world, was absent in philosophy. This observation was encountered in different ways. Habermas developed his Theory of Communicative Action [Ha81] in order to provide such a concept (see Section 3.1). Habermas' theory had strong influence on the way FCA was developed. Computer scientists, on the other hand, extended the definition of the concept 'ontology' — and adapted it in a straightforward manner directly to their own purposes (which led to many controversies with philosophers). Most popular in computer science is nowadays the definition of T. Gruber, who considers ontologies as "formal, explicit specification of a shared conceptualization" [Gr94]. A 'conceptualization' refers to an abstract model of some phenomenon in the world by identifying the relevant concept of that phenomenon. 'Explicit' means that

[7] www.fzbw.tu-darmstadt.de

the types of concepts used and the constraints on their use are explicitly defined. 'Formal' refers to the fact that the ontology should be machine understandable (which excludes for instance natural language). 'Shared' reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

In practice, the two approaches are not far from each other. Both FCA and Ontology Engineering emphasize the importance of an inter-subjective agreement about the conceptualization, and both claim the need of a formal specification of the model. The main difference is that, in terms of ISO 704 (see Section 3.2), FCA works mainly on the concept level, while Ontology Engineering works mainly on the representation level. I. e., FCA considers extensional and intensional aspects as equal, while Ontology Engineering emphasizes on the intensional part. As already argued in Section 3.2, these views should be understood as complementary rather than competitive. We suggest thus to integrate Formal Concept Analysis and Ontology Engineering in one unified framework. Establishing this framework and working on its details are interesting topics for future research.

5 Conceptual Knowledge Discovery and Processing

In this section, we present *Conceptual Knowledge Processing (CKP)* which arose as an extension of FCA taking into account more explicitly Davis *et al*'s fourth and fifth principles; and argue why it is a reasonable choice for a framework unifying FCA and Ontology Engineering.

5.1 Conceptual Knowledge Processing

Conceptual Knowledge Processing (CKP) has as its overall aim supporting human communication and argumentation to establish inter-subjectively assured knowledge. As a computer science theory, the task of CKP is thus to provide concept–based methods and tools for acquiring, representing, and reasoning with knowledge, and for making it available for communication purposes. We analyze how FCA (with its recent extensions) fulfills this task and how it can be complemented by Ontology Engineering in the aim of supporting Conceptual Knowledge Processing. We consider the following four categories of knowledge processing: *knowledge acquisition, knowledge representation, knowledge inference*, and *knowledge communication* [LuW91].[8] We will focus on technical aspects; a reflection of the philosophical foundations of CKP can be found in [Wi94] and [Wi99].

[8] These categories have been addressed in variations by many authors, for instance by A. M. Kleinhans [Kl89] as knowledge acquisition, knowledge storage, knowledge manipulation, and knowledge distribution; or in a more fine-grained categorization by G. Probst, S. Raub, and K. Romhardt in [PRR99].

Knowledge Acquisition. Knowledge Acquisition techniques (in the broader sense) can roughly be categorized in two classes: those which aim at acquiring knowledge from humans (i. e., knowledge acquisition in the narrower sense), and those which acquire knowledge out of some data (e. g., documents) in which the knowledge is encoded. As we will argue below, we do not see the two classes far from each other. The latter class is subject of the research domains Machine Learning and (more recently) Knowledge Discovery. This article has a certain focus on the second class, and therefore devotes the entire next subsection to it. There we analyze the roles of Conceptual Knowledge Discovery and of Ontology Learning.

As for the techniques for knowledge acquisition from humans, the most prominent representative within FCA is B. Ganter's *Attribute Exploration* [Ga87] (see also [GW99a]). It addresses the problem of a context where the object set is not completely known a priori, or too large to be completely listed. In an interactive, iterative approach, the user has either to accept a suggested implication between the attributes (i. e., she excludes potential objects) or to provide a counter-example (i. e., she provides a (typical) object) until the concept lattice is completely determined. Concept Exploration extends this approach to situations where both the object set and the attribute set of the context are not completely known a priori or too large [KM88,St97]. An overview over interactive knowledge acquisition techniques based on FCA can be found in [St96c]. Also more informal knowledge acquisition settings within FCA aim at the specification of the formal context. In a typical data analysis scenario, the first step is to establish a formal context in cooperation with the user(s). Based on the insights gained by the resulting concept lattice, the context can be refined and modified in subsequent feedback loops.

Ontology Engineering in its turn has its roots in the Knowledge Acquisition community. From there, it brings along methodologies for knowledge acquisition, as for instance CommonKADS [Sc+00], which is currently instantiated for ontologies in the OTK ontology development framework [SSSS01]. Recent knowledge acquisition approaches within Ontology Engineering can be classified in two groups: ontology learning and instance learning (information extraction). The first deals with learning the ontology itself (i. e., the intensional aspect), and the second with learning the assignment of instances to the concepts and relations (i. e., the extensional aspect). The first group, ontology learning, uses techniques from natural language processing and data mining (e. g., clustering and association rules) to support the interactive process of building the ontology [Mä02]. Ontology learning follows the paradigm of balanced cooperative modeling [Mo93] which we will discuss in the Section 5.2. Ontology learning comprises the task of merging ontologies from different sources. In Section 6.2, we discuss a FCA-based technique for this task. A comprehensive overview over ontology learning is given in [Mä02]. The second group, instance learning, includes techniques for annotation and authoring. Annotation, now one of the major techniques for creating meta-data in the world-wide web, aims at attaching ontology instantiations to web pages [HSM01]. Authoring supports this attachment on-line during the construction of a web page [HS02]. These approaches use for instance Information Extraction and Document Management techniques, as well as inference and crawler mechanisms.

Like FCA, Ontology Engineering emphasizes on the importance of agreeing among the domain experts on a shared understanding of the domain. One difference is that most of the Ontology Engineering approaches base the interactive knowledge acquisition process on heuristics which allow for more flexibility than FCA approaches. In general one can conclude that Ontology Engineering provides more comprehensive support for the more informal aspects of knowledge acquisition and complements thus well with the more structure-oriented techniques of FCA which come along with stronger semantics.

Knowledge Representation. Knowledge representation with FCA has already been the overall theme of Section 3. Here we focus on its relationship to Ontology Engineering.

The choice of the formalism for representing an ontology directly influences the methods and tools to be applied; there is no language–neutral Ontology Engineering. Ontologies are described in different formalisms (e. g., description logics, conceptual graphs, frame logic), depending on the task to be solved (and on the history of the researcher working on it). As argued in Section 3.2, these formalisms complement well with FCA, and first steps have been made to set up links between the underlying theories. These links have to be strengthened and are to be exploited for establishing a comprehensive Conceptual Knowledge Processing environment. From the FCA perspective, this means to extend the scope from strongly structured to semi–structured and even unstructured data, allowing to tackle more complex tasks as, for instance, in the Semantic Web.

Knowledge Inference. The second important thread in CKP today is, beside Conceptual Knowledge Discovery, the development of *Contextual Logic* [Wi96] [Wi00]. Contextual Logic aims at restructuring mathematical logic, following Hentig's restructuring program, in order to overcome deficiencies of predicate logic for knowledge representation [Pr00]. It is based on the elementary doctrines of concepts, judgments, and conclusions as discussed in classical philosophical logic. In this framework, FCA is considered as a theory for concepts, while Conceptual Graphs are building blocks for a theory for judgments and conclusions. Due to space restrictions, Contextual Logic will not be presented in detail in this article. The interested reader is referred to [Wi96,Wi97,Pr98,Pr00,Wi00].

Davis *et al* suggest to analyze two sets of inferences for a given knowledge representation: the set of inferences the representation sanctions, and the set of inferences it recommends. As known from other mathematics–based logics, Contextual Logic currently provides a sound and complete set of inferences, i. e., a set of inferences the representation sanctions. The choice of the inferences to be applied is left to the user; she is supported in this task by graphical user interfaces [EGSW00].

Ontology Engineering tools in general make use of sanctioned inferences, too, for instance for checking the consistency of the ontology, and for deriving knowledge which is not explicitly encoded. As there is no language–neutral representation of an ontology, each Ontology Engineering tool has to provide an implementation of an inference mechanism applicable to the language it uses. Additionally to the set of sanctioned inferences, Ontology Engineering tools often make extensive use of heuristics (as for instance in the merging tools listed in Section 6.2), which can be seen as implementations of sets of recommended inferences. A tighter interweaving of heuristics–based approaches with FCA and Contextual Logic is an interesting topic for future research.

Knowledge Communication. For Formal Concept Analysis, the importance of knowledge communication has already been discussed in Section 3.1. This aspect has been the driving force for the development of several tools, e. g., ConImp [Bu87,Bu00], GALOIS [CR93], the management system TOSCANA for Conceptual Information Systems [VW95] with various extentions (e. g., [SW97,SW98] [MSW99,St00,EGSW00,HS01,To01], see also Section 8.1 and 8.2) and the analysis tool CERNATO[9].

Ontologies also have as primary focus the support of human (and human–computer) communication. They are applied for instance for community building [Sta+00], for knowledge management [ABHKS98,SMS00,SSSS01,ASSS00], and in the Semantic Web [BHL01]. The Semantic Web aims at providing automated Web services based on formal knowledge representations. In this scenario, ontologies are used for instance in semantics–based portals [SMSSS01,StaM01,JKN01] and for the communication of (software) agents [Hd01].

Systems like WAVE for web navigation [KN95], the expert system MCRDR [RC97], the RFCA system for browsing rental advertisements on the WWW [CE01] or the Conceptual Email Manager (which is discussed in Section 8.3) are first prototypes integrating both FCA and ontologies. The next step will be to establish interfaces between the two research and software projects 'Tockit — Framework for Conceptual Knowledge Processing'[10] and 'KAON — Karlsruhe Ontology and Semantic Web Tool Suite'[11] in order to obtain a large, stable platform for future developments.

5.2 Conceptual Knowledge Discovery

The aim of *Knowledge Discovery in Databases* (*KDD*) is to support human analysts in the overall process of discovering valid, implicit, potentially useful and ultimately understandable information in databases. The volume "Advances in Knowledge Discovery and Data Mining" [FPSU96] emphasizes that this iterative and interactive process between a human and a database may strongly

[9] http://www.navicon.de/deutsch/sit_f.htm
[10] http://tockit.sourceforge.net/
[11] http://kaon.semanticweb.org/

involve background knowledge of the analyzing domain expert.[12] In particular, R. S. Brachman and T. Anand [BA96] argue in favor of a more human-centered approach to knowledge discovery ("data archeology", [BST+93]) referring to the constitutive character of human interpretation for the discovery of knowledge and stressing the complex, interactive process of KDD as being led by human thought. Following Brachman and Anand, *Conceptual Knowledge Discovery (CKDD)* pursues a human-centered approach to KDD based on a comprehensive notion of knowledge as a part of human thought and argumentation [SWW98,HSWW00]. This view leads to a modified definition of what knowledge discovery is: we understand (conceptual) knowledge discovery as "information discovery combined with knowledge creation where the combination is given by turning discovered information into created knowledge" [Wi01b]. A more detailed discussion of this understanding along a list of requirements for knowledge discovery environments provided in [BA96] can be found in [SWW98].

In most applications, classical data analysis and decision support facilities (for instance Online Analytical Processing (OLAP) or statistical packages) are already present when data mining tools are added to the knowledge discovery support environment. For supporting the analyst in the overall process of human-centered knowledge discovery, both decision support and data mining tools should provide a homogeneous environment. In particular, this shows the need of a unified knowledge representation. In Section 3, we argued for formal contexts and concept lattices as such a unified knowledge representation. In Sections 6 and 7, we will discuss some applications. Further CKDD applications are presented in [St98], [St99c], [HSWW00], and [Du+01].

The human–centered approach of CKDD indicates the need to distribute the work between data mining algorithms on the one hand and the user on the other hand. The same observation has been made for the combination of machine learning and knowledge acquisition by K. Morik and led to the approach of *balanced cooperation* [Mo93]. Ontology Learning, the knowledge discovery part of Ontology Engineering, adopted this paradigm: A. Mädche considers the process of Ontology Learning as a semi-automatic process with human intervention, since completely automatic knowledge acquisition is an unrealistic vision (today) [Mä02, p. 52]. The approach allows the integration of a multitude of disciplines (e. g., machine learning, natural language processing, human–computer interaction) in order to facilitate the semi–automatic construction of ontologies.

Instance learning, as discussed in the previous subsection, is today more based on user-centered, interactive techniques (that is why we discussed it under the heading 'knowledge acquisition' above, and not here). However, we expect that instance learning will make a more extensive use of data mining techniques in the near future, converging hence also to the paradigm of balanced cooperation.

As discussed above, we want to integrate Ontology Engineering into Conceptual Knowledge Processing. For Conceptual Knowledge Discovery, this means

[12] Following [FPSU96], we understand KDD as the overall discovering process; while *data mining* is considered as one step of KDD, namely the application of algorithms for extracting patterns from the data.

that Ontology Learning, Instance Learning, and FCA–based knowledge discovery should be brought together. Our vision for future research is to interweave these approaches, and to apply them for concept-based knowledge discovery. This is especially promising in the upcoming Semantic Web, where first steps towards *Semantic Web Mining* have been done [SHB01].

In the next three sections, we present three lines of research which exemplarily demonstrate the extension of FCA to Conceptual Knowledge Processing and its change of orientation towards computer science. The aim is to show how this change of orientation inspired research within the whole bandwidth from theory to applications. The topics are

- the use of FCA as a conceptual clustering technique and its application for supporting the merging of ontologies,
- the efficient computation of association rules, and the structuring and reduction of the results,
- and the visualization and the management of conceptual hierarchies/ontologies, and its application in an email management system.

6 Conceptual Clustering and Ontology Merging

Cluster Analysis comprises a set of unsupervised machine learning techniques which split sets of objects into clusters (subsets) such that objects within a cluster are as similar as possible while objects from different clusters are as different as possible. Conceptual Clustering techniques additionally aim at determining not only clusters — i. e., concept extensions — but to provide at the same time intensional descriptions of these extensions [Mi80,WMJ00]. This aim fits well with the understanding of concepts as given in Section 4. Thus it is natural to consider FCA as a framework for conceptual clustering (see also [StrW93,CR93,MG95]). In this section, we discuss one particular way of conceptual clustering with FCA, namely *iceberg concept lattices.* Iceberg concept lattices and the algorithm Titanic for their computation are described in detail [STBPL02].

We apply iceberg concept lattices in Ontology Engineering. We use them to support the ontology engineer in merging ontologies. This structure-oriented method, called FCA–Merge, is described in detail in [SM01].

6.1 Computing Iceberg Concept Lattices with Titanic

Compared to 'usual' clustering, conceptual clustering techniques pay their added value (the intensional description) with increased computation time. In FCA, there exist basically three ways to overcome this problem: local focusing (e. g., [CR93]), vertical reduction by conceptual scaling (see Section 2), and horizontal reduction. All three ways reduce not only the computation time, but also the amount of information presented to the user.

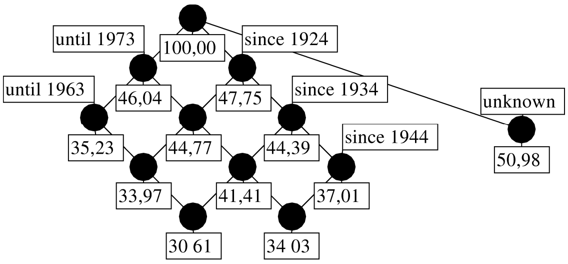

Fig. 5. Iceberg concept lattice for customers clustered by their year of birth.

Iceberg concept lattices are a horizontal approach to reduce the amount of information (and the computation time) of a concept lattice. They have first been mentioned as 'frequent concept lattices' in [St99b], and are discussed in detail in [STBPL02], where also an efficient algorithm for their computation is presented, the TITANIC algorithm. Iceberg concept lattices show only the topmost part of a concept lattice. In Figure 5, for instance, the customers of a warehouse in Zurich, Switzerland, are clustered according to their year of birth. The minimum support threshold is set to 0.3, i. e., all concepts whose extents do not comprise at least 30 % of all customers, are pruned. Instead of the objects names, the diagram displays the support of the concepts, i. e., its relative size compared to the total number of objects.

Iceberg concept lattices have different uses in KDD: as conceptual clustering tool, as a visualization method — especially for *very large* databases —, as a condensed representation of frequent itemsets, as a base of association rules, and as a visualization technique for association rules. In comparison to other conceptual clustering approaches, iceberg concept lattices have structural properties which can be stated explicitly: they do not depend on diverse parameters (except the minimum support threshold) whose semantics are often difficult to interpret, nor on the order in which the input is presented to the algorithm, nor on any particularities of the implementation. Another distinction to other hierarchical clustering results is that they allow for multiple hierarchies (and not only for trees), so that all potentially interesting specialization paths are contained in the resulting hierarchy.

The TITANIC algorithm addresses the problem of computing (iceberg) concept lattices from a data mining viewpoint by using a level-wise approach [AS94,MT97]. This allows to decrease significantly the computation time compared to the reference algorithm Next-Closure [Ga87]. TITANIC can be applied to a broad class of problems: Computing arbitrary closure systems when the closure operator comes along with a so-called weight function. Weight functions appear naturally in a variety of applications, including association rule mining, functional dependencies in databases, conceptual clustering, transformation of class hierarchies in object-oriented languages, configuration space analysis in software re-engineering, and ontology engineering.

6.2 FCA–Merge: Bottom-Up Merging of Ontologies

Ontologies have been established for knowledge sharing and are widely used as a means for conceptually structuring domains of interest. With the growing usage of ontologies, the problem of overlapping knowledge in a common domain occurs more often and becomes critical. Domain-specific ontologies are modeled by multiple authors in multiple settings. These ontologies lay the foundation for building new domain-specific ontologies in similar domains by assembling and extending multiple ontologies from repositories.

The process of *ontology merging* takes as input two (or more) source ontologies and returns a merged ontology based on the given source ontologies. Manual ontology merging using conventional editing tools without support is difficult, labor intensive and error prone. Therefore, several systems and frameworks for supporting the knowledge engineer in the ontology merging task have recently been proposed [Ho98,Ch00,NM00,MFRW00]. The approaches rely on syntactic and semantic matching heuristics which are derived from the behavior of ontology engineers when confronted with the task of merging ontologies, i. e., human behavior is simulated. Although some of them locally use different kinds of logics for comparisons, these approaches do not offer a structural description of the global merging process.

In [SM01], the method FCA–Merge for merging ontologies following a bottom-up approach and offering a global structural description of the merging process is presented. For the source ontologies, it extracts instances from a given set of domain-specific text documents by applying natural language processing techniques. Based on the extracted instances we use the Titanic algorithm to derive a concept lattice. The concept lattice provides a conceptual clustering of the concepts of the source ontologies. It is explored and interactively transformed to the merged ontology by the ontology engineer.

Certainly, high quality results of the merging process will always need a human involved who is able to make judgments based on background knowledge, social conventions, and purposes. Thus, all merging approaches aim at supporting the knowledge engineer, and not at replacing him. Our approach differs from the related work stated above in that it provides, for one part of the merging process, an algorithm with a well-defined description of the output in terms of the input. If the knowledge engineer commits to this description, he is guaranteed to obtain the expected results. FCA–Merge may of course also be included in any heuristic-based approach as a — reliable — building block.

7 Computation and Reduction of Association Rules

One of the core tasks of *Knowledge Discovery in Databases* (*KDD*) is the mining of association rules (conditional implications). *Association rules* are statements of the type '67 % of the customers buying cereals and sugar also buy milk (where 7% of all customers buy all three items)'. The task of mining association rules is to determine all rules whose *confidences* (67 % in the example) and *supports* (7 %

in the example) are above user-defined thresholds. Since the problem was stated [AIS93], various approaches have been proposed for an increased efficiency of rule discovery in very large databases [AS94,BA98,BMUT97,PBTL99b,PBTL99a].

Frequent patterns are those subsets of the attribute set (set of items) whose support is above a certain threshold. They are the result of the (computationally expensive) first step of the typical two-step approach for mining association rules. In the next subsection, we discuss a new algorithm for this first step, called Pascal. It is described in more detail in [BTPSL00].

In Subsection 7.2, we discuss how to reduce the number of extracted association rules *without losing any information*, based on mathematical properties of closure systems. This is described in more detail in [STBPL01a].

7.1 Mining Frequent Patterns with Counting Inference

The problem of mining frequent patterns arose first as a sub-problem of mining association rules, but it then turned out to be present in a variety of problems [HPY00]: mining sequential patterns [AS95], episodes [Ma97], association rules [AS94], correlations [BMS97,SBM98], multi-dimensional patterns [KHC97] [LSW97], maximal patterns [BA98,ZPOL97,LK98], closed patterns [BPTSL00] [PBTL99a,PBTL99b,PHM00]. Since the complexity of this problem is exponential in the size of the binary database input relation and since the relation has to be scanned several times during the process, efficient algorithms for mining frequent patterns are required.

Three approaches have been proposed in the literature for mining frequent patterns: All of them traverse iteratively the set of all patterns in a levelwise manner. The first approach follows the classical Apriori [AS94] algorithm (e. g., [BMUT97,PCY95,SON95,To96]). The second combines this idea with extracting maximal frequent patterns (e. g., [BA98]), while the third approach (e. g., [PBTL99a]), combines it with structural information provided by FCA theory. While all of the algorithms of the first two approaches have to determine the supports of *all* frequent patterns and of some infrequent ones *in the database*, the third approach allows to derive a portion of them from already known supports.

Our Pascal algorithm is an effective and simple optimization of the Apriori algorithm. It belongs to the third category discussed above, and is a cousin of Titanic. It applies FCA results to solve the data mining task of computing all frequent patterns (and not only the closed ones as Titanic does). Pascal is based on *pattern counting inference* that relies on *key patterns* (or *minimal generators*). A key pattern is a minimal pattern of an equivalence class gathering all patterns that have the same objects. Our optimization is based on the fact that key patterns form an order ideal in the powerset of the attribute set, which is a necessary condition for applicability of the Apriori pruning strategy. The pattern counting inference allows to determine the supports of the key patterns in the database only. The supports of all other frequent patterns are derived from the frequent key patterns without accessing the database. This allows to reduce, at each database pass, the number of patterns considered, and, even more important, to reduce the number of passes in total. As shown by experiments, the efficiency gain compared to Apriori is up to one order of magnitude on correlated data.

7.2 Intelligent Structuring and Reducing Association Rules by Formal Concept Analysis

Following Davis *et al*'s first principle, fully taking advantage of discovered association rules means providing capabilities to handle them. The problem is especially critical when collected data is highly correlated or dense, like in statistical databases [BMUT97]. For instance, when applied to a census dataset of 10,000 objects — each of which is characterized by values of 73 attributes — experiments result in more then 2,000,000 rules with support and confidence greater than or equal 90%. Thus the question arises: How can long lists of association rules be reduced in size?

Approaches addressing the described issue provide users with mechanisms for filtering rules, for instance by user defined templates [BP97,KMRTV94], Boolean [NLP98,SVA97] or SQL-like [MPC96] operators or by introducing further measures of "usefulness" [BA99]; or they attempt to minimize the number of extracted rules a priori by using information about taxonomies [HF95,HMWG98] [SA95] or by applying statistical measures like Pearson's correlation or the χ^2-test [BMS97]. All these approaches have in common that they lose some information.

Our approach described in [STBPL01a], on the other hand, allows us to significantly reduce the number of rules without losing any information. We extract only a subset of all association rules, called *basis*, from which all other rules can be derived. This approach is orthogonal to the ones mentioned above and can be combined with them. We use two complementary bases, extending results of Duquenne and Guigues ([DG86], cf. also [GW99a]) and Luxenburger [Lu91,Lu93]. The former have studied bases for association rules with 100 % confidence, and the latter association rules with less than 100 % confidence. We adopt their results to association rules (where also the support is considered) and provide algorithms for computing the new bases.

The *Duquenne-Guigues basis for exact association rules* consists of rules where the premise is a so-called frequent pseudo-intent, and the conclusion a related frequent concept intent. The *Luxenburger basis for approximate association rules* contains only rules where premise and conclusion are frequent concept intents of concepts being neighbors in the concept lattice. To give an impression, the Luxenburger basis for the MUSHROOMS database[13] with a minimum support of 70 % and a minimum confidence of 95 % is shown in Figure 6. We have proven that these bases are minimal with respect to a certain set of inference rules.

In [BPTSL00], we have presented another pair of bases, which provide rules with minimal antecedents and maximal consequents. Compared to the results discussed here, they have the disadvantage of a higher total number of rules. For the approximate rules, M. Zaki has presented similar results in [Za00]. However,

[13] ftp://ftp.ics.uci.edu/~cmerz/mldb.tar.Z

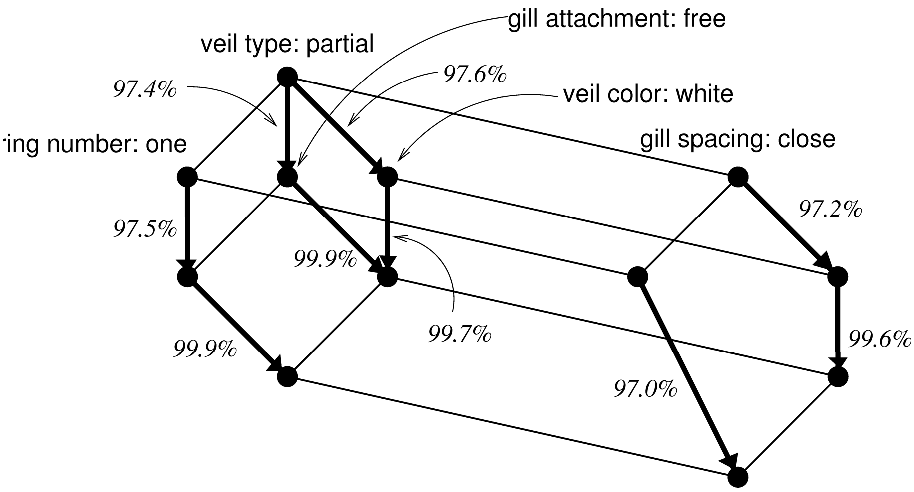

Fig. 6. Luxenburger basis for the Mushrooms database

he does not provide inference rules for support and confidence derivation, does not discuss minimality of his results, and does not provide algorithms for the computation of the bases.

For the computation of the bases, we follow an approach in two steps. In the first step, we compute all frequent patterns, and determine all concept intents among them, using the PASCAL algorithm described in the previous subsection. In the second step, we derive the bases for the association rules. Experiments show that, by exploiting the lattice structure, we are able to reduce the number of rules by up to 2.5 orders of magnitude without loosing any information; and to significantly speed up the computation, especially for strongly correlated data or when the minimum support is low.

8 Visualization and Management of Conceptual Hierarchies

An important problem in the application of FCA is the size of the concept lattices. In the worst case, their size is exponential in the size of the formal context. Hence methods for managing large conceptual hierarchies and for visualizing (parts of) them are needed.

One solution for visualizing a part of the hierarchy has already been discussed in Section 6.1. Another popular approach to overcome this problem — which is orthogonal to iceberg concept lattices — is conceptual scaling combined with nested line diagrams as described in Section 2. *Local scaling* is a technique for further reducing the complexity of those nested line diagrams. It will be discussed in the next subsection. Details of the construction are given in [St96b].

In very large applications, conceptual scaling alone is not sufficient to support the user in navigating through the hierarchy, since the total number of conceptual scales becomes too large. The need for navigation support on this meta-level arises. In Subsection 8.2, we discuss the use of a hierarchy on the set of conceptual scales to support this navigation. More details are given in [St99a].

Both local scaling and a hierarchy on the conceptual scales are used in the Conceptual Email Manager CEM. The purpose of this email manager is to improve the retrieval facilities of standard email management systems by exploiting the fact that concept lattices allow for multiple hierarchies. CEM is discussed in Subsection 8.3, and described in detail in [CS00].

8.1 Local Scaling in Conceptual Data Systems

Nested line diagrams may become unnecessarily large, if the second scale does not differentiate all of the concepts of the first scale. For instance, in Figure 3, one may wonder if the concepts of the outer scale labeled by 'Europe' and 'Asia Pacific' really need to be enlarged. Intuitively, one may just collapse them. *Local scaling*, as discussed below, reduces the complexity of the nested line diagram, and provides at the same time a clear semantic.

It is a known result that the concept lattice of the derived context of a set of conceptual scales can be embedded (as a join-semi-lattice) in the direct product of their concept lattices. This embedding was used to generate the nested line diagram in Figure 3. The basic idea of local scaling is to find a construction smaller than the direct product such that the concept lattice of the derived context can still be embedded into it (again as a join-semi-lattice).

Local scaling is inspired by so-called local direct products [Ge88], a generalization of A. Day's doubling construction for solving the word problem in free lattices [Da92]. In [St96a], the detailed construction of local scaling is described. Its semantics is provided, and the correctness is proved. The main condition for obtaining unambiguous diagrams turns out to be that the set of concepts of the outer scale to be enlarged must be convex. For the diagram in Figure 3, this means that the two concepts labeled by 'Europe' and 'Asia Pacific' have to be enlarged, as they are positioned between two concepts which have to be enlarged as well. Hence in this case we cannot reduce the visualization by local scaling.

Based on local scaling, one can provide different techniques for interactively exploring the data: By iteratively enlarging relevant concepts, one can simulate a 'conceptual magnifying glass', and one can support 'parallel zooming' into concepts of the outer scale. Local scaling is implemented in the Conceptual Email Manager discussed below (see Figure 7).

8.2 Hierarchies of Conceptual Scales

In large applications, the number of conceptual scales may be so large that navigation support is needed on this meta-level, too. By introducing new *higher level attributes* and a taxonomy on these new attributes, one can derive new, hierarchically ordered *higher level scales.* We obtain a cascading hierarchy of

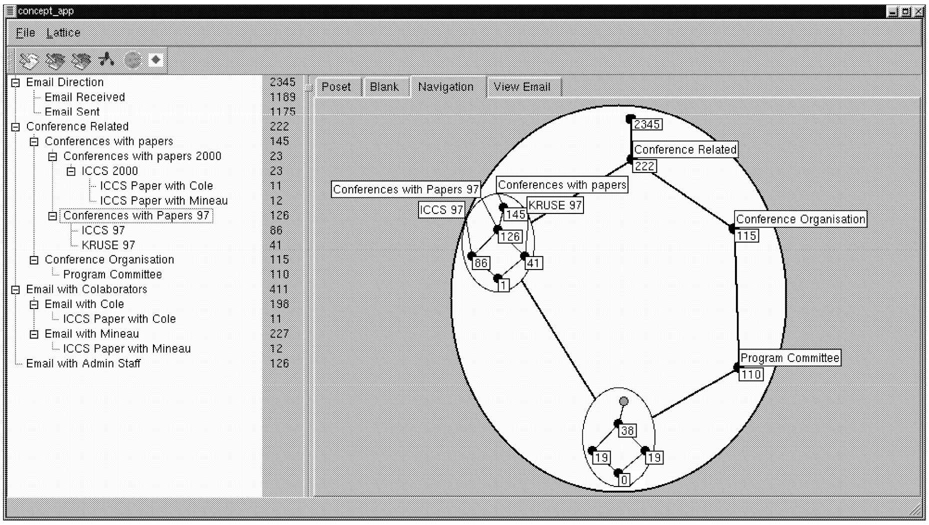

Fig. 7. Screenshot of the Conceptual Email Manager. Here it uses local scaling to combine the two scales 'Conference Related' and 'Conferences with papers'.

conceptual scales with increasing granularity. Higher level scales (which can be derived automatically from the taxonomy) provide information about the data on a more general level. They allow to observe global 'cross-scale' relationships that cannot be recognized easily otherwise. Hence higher level scales are an interesting technique for knowledge discovery while they allow at the same time a drill-down to the original data as known from Online-Analytical Processing (OLAP; cf. also to [St00]). The technique of local scaling can be used for visualizing conceptual hierarchies located on different levels. Its adaption to hierarchies of scales, called *nested scaling*, significantly reduces the complexity of the visualization.

The construction of higher level scales can also be used to support navigation in ontologies. Therefore, a hierarchy of conceptual scales has to be derived from the ontology. Instead of a bottom-up approach as discussed above, one can here apply a top-down approach. Using the subconcept-superconcept-relation of the ontology, one obtains automatically a conceptual scale for each concept of the ontology by choosing as attributes of the scale its immediate subconcepts in the ontology.

In [St99a], hierarchies of concepts have been applied for enhancing the navigation support for information retrieval in the library of the Center of Interdisciplinary Research of Darmstadt University of Technology (see [RW00]). They are also implemented in the Conceptual Email Manager which is discussed next.

8.3 CEM – A Conceptual Email Manager

The way standard email management systems store mails is directly derived from the tree structure of filing cabinets and file management systems. This has the advantage that trees have a simple structure which can easily be explained to novice users. The disadvantage is that at the moment of storing an email the user already has to foresee the way she is going to retrieve the mail later. The tree structure forces her to decide at that moment which criteria to consider as primary and which as secondary. For instance, when storing an email regarding the organization of a conference, one has to decide whether to organize one's directories like `mineau/iccs2000/program_commitee` or like `conferences/iccs/iccs2000/organisation/mineau`. This problem arises especially if a user communicates with overlapping communities on different topics.

In [CS00], we introduce the *Conceptual Email Manager CEM*. It uses a formal context as its structure for storing email rather than a tree. The objects are all emails stored by the system, and the attributes are catchwords like 'conferences', 'mineau', and 'organisation'. This allows the user to retrieve emails via a concept lattice following different paths. For the example above this means that one need not decide which of the two paths to use for storing. For retrieving the mail later, one can consider any combination of the catchwords in the two paths. This is made possible by using the richly structured concept lattice as search space. The Conceptual Email Manager uses a hierarchy on the conceptual scales as discussed above, and uses local scaling for visualizing the search space. Figure 7 shows a screen-shot of the Conceptual Email Manager. The use of the Conceptual Email Manager for Conceptual Knowledge Discovery is discussed in [CS00].

Our approach is related to the use of *virtual folders* in the program View Mail (VM) [Jo99], which are collections of email documents retrieved in response to a query. It is also related to the library information system implemented in the Center of Interdisciplinary Studies at Darmstadt University of Technology [RW00] based on the management system TOSCANA for Conceptual Information Systems [VW95]. The retrieval components of both our system and the library system provide basically the same functionality. The difference lies in the enhanced support for the user maintaining and updating the email collection: while in the library system maintenance is allowed only to the librarian and/or a knowledge engineer, in an email management system storing emails is an essential and often used feature which requires some semi-automatic support for a (relatively) untrained user.

9 Outlook

In this article, we have discussed the turn of FCA towards computer science. We have analyzed why FCA is considered as a knowledge representation method within computer science, and how and why mathematics–based FCA researchers became attracted by computer science. We presented Conceptual Knowledge Processing and Conceptual Knowledge Discovery as steps in that development, and argued for a future integration with Ontology Engineering. The turn of orientation of FCA has provided interesting results, as the discussed research topics and applications show. We strongly believe that there remains a huge scientific potential in the exploitation of bringing together mathematical–structural results (especially from FCA) and procedural aspects, which will further enhance the state of the art in computer science.

Acknowledgements

I am grateful to Susanne Prediger for intensive discussions about the vision described in this article, and to my colleagues for pointing out specific relationships to Ontology Engineering.

References

[ABHKS98] A. Abecker, A. Bernandi, K. Hinkelmann, O. Kühn, M. Sintek: Towards a technology for organizational memories. *IEEE Intelligent Systems and Their Applications* **13**(3), 1998, 40–48

[AIS93] R. Agrawal, T. Imielinski, A. Swami: Mining association rules between sets of items in large databases. *Proc. SIGMOD Conf.*, 1993, 207–216

[AS94] R. Agrawal, R. Srikant. Fast algorithms for mining association rules. *Proc. of the 20th Int'l Conf. on Very Large Data Bases (VLDB)*, 1994, 478–499 (Expanded version in IBM Report RJ9839)

[AS95] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. of the 11th Int'l Conf. on Data Engineering (ICDE)*, March 1995, 3–14

[ASSS00] J. Angele, H.–P. Schnurr, S. Staab, R. Studer: The times they are a-changin' — the corporate history analyzer. In: D. Mahling. U. Reimer (eds.): *Proc. 3rd Intl. Conf. on Practical Aspects of Knowledge Management*, Basel, October 2000. www.research.swisslife.ch/pakm2000

[AN85] A. Arnaud, P. Nicole: *La logique ou l'Art de penser.* Abraham Wolfgang, Amsterdam 1685. German translation: *Die Logik oder die Kunst des Denkens.* 2. Aufl. Wissenschaftliche Buchgesellschaft, Darmstadt 1994

[Ba95] F. Baader: Computing a minimal representation of the subsumption lattice of all conjunctions of concept defined in a terminology. In: G. Ellis, R. A. Levinson, A. Fall, V. Dahl (eds.): *Proc. Intl. KRUSE Symposium*, August 11–13, 1995, UCSC, Santa Cruz 1995, 168–178

[BP97] E. Baralis, G. Psaila: Designing templates for mining association rules. *Journal of Intelligent Information Systems* **9**(1), 1997, 7–32

[BPTSL00] Y. Bastide, N. Pasquier, R. Taouil, G. Stumme, L. Lakhal: Mining Minimal Non-Redundant Association Rules Using Frequent Closed Itemsets. In: J. Lloyd, V. Dahl, U. Furbach, M. Kerber, K.–K. Lau, C. Palamidessi, L. M. Pereira, Y. Sagiv, P. J. Stuckey (eds.): *Computational Logic — CL 2000.* Proc. CL '00. LNAI **1861**, Springer, Heidelberg 2000, 972–986

[BM70] M. Barbut, B. Monjardet: *Ordre et classification, Algèbre et Combinatoire.* 2 tomes. Paris, Hachette 1970

[BTPSL00] Y. Bastide, R. Taouil, N. Pasquier, G. Stumme, L. Lakhal: Mining Frequent Patterns with Counting Inference. *SIGKDD Explorations* **2**(2), Special Issue on Scalable Algorithms, 2000, 71–80

[BA98] R. J. Bayardo: Efficiently mining long patterns from databases. *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, June 1998, 85–93

[BA99] R. Bayardo and R. Agrawal: Mining the most interesting rules. In *Proc. of the 5th Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, Aug. 1999, 145–154

[BHL01] T. Berners-Lee, J. Hendler, O. Lassila: The Semantic Web. *Scientific American* **284**(5), May 2001, 34–43

[Bi40] G. Birkhoff: *Lattice Theory.* 1st edition. Amer. Math. Soc. Coll. Publ. **25**, Providence, R. I., 1940

[Bra79] R. J. Brachman: On the epistemological status of semantic networks. In: N. V. Findler (ed.): *Associative Networks: Representation and Use of Knowledge by Computers.* Academic Press, New Work 1979, 3–50

[BA96] R. J. Brachman, T. Anand: The process of Knowledge Discovery in Databases. In: U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.): *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press, Cambridge 1996, 37–57

[BST+93] R. J. Brachman, P. G. Selfridge, L. G. Terveen, B. Altman, A. Borgida, F. Helper, T. Krk, A. Lazar, D. L. McGuiness, L. A. Resnick: Integrated support for data archeology. *Intl. J. of Intelligent and Cooperative Information Systems* **2** (1993), 159–185

[BMUT97] S. Brin, R. Motwani, J. D. Ullman, S. Tsur: Dynamic itemset counting and implication rules for market basket data. *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, May 1997, 255–264

[BMS97] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlation. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, May 1997, 265–276

[Bru76] W. Brugger: *Philosophisches Wörterbuch* Herder, Freiburg 1976

[Bu87] P. Burmeister: Programm zur Formalen Begriffsanalyse einwertiger Kontexte. TH Darmstadt 1987

[Bu00] P. Burmeister: ConImp — Ein Programm zur Formalen Begriffsanalyse. In: G. Stumme, R. Wille (eds.): *Begriffliche Wissensverarbeitung — Methoden und Anwendungen.* Springer, Heidelberg 2000, 25–56

[CR93] C. Carpineto, G. Romano: GALOIS: An Order-Theoretic Approach to Conceptual Clustering. *Machine Learning.* Proc. ICML 1993, Morgan Kaufmann Prublishers 1993, 33–40

[Ch00] H. Chalupsky: OntoMorph: A translation system for symbolic knowledge. *Proc. 7th Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'2000)*, Breckenridge, Colorado, USA, April 2000,471–482

[CE01] R. Cole, P. Eklund: Browsing Semi-Structured Web Texts Using Formal Concept Analysis. In: H. Delugach, G. Stumme (Eds.): *Conceptual Structures: Broadening the Base.* Proc. ICCS '01. LNAI **2120**, Springer, Heidelberg 2001, 319–332

[CES00] R. Cole, P. Eklund, G. Stumme: CEM — A Program for Visualization and Discovery in Email. In: D. A. Zighed, J. Komorowski, J. Zytkow (eds.): *Principles of Data Mining and Knowledge Discovery.* Proc. PKDD 2000. LNAI **1910**, Springer, Heidelberg–Berlin 2000, 367–374

[CS00] R. Cole, G. Stumme: CEM – a conceptual email manager. In: B. Ganter, G. W. Mineau (eds.): *Conceptual Structures: Logical, Linguistic, and Computational Issues.* Proc. ICCS '00. LNAI **1867**. Springer, Heidelberg 2000, 438–452

[DSS93] R. Davis, H. Shrobe, P. Szolovits: What is a knowledge representation? *AI Magazine* **14:1** (1993), 17–33.

[Da92] A. Day: Doubling constructions in lattice theory. *Canad. J. Math.* **44**, 1992, 252–269

[DIN2330] Deutsches Institut für Normung: *Begriffe und Benennungen – Allgemeine Grundsätze.* DIN 2330. 1993

[DIN2331] Deutsches Institut für Normung: *Begriffssysteme und ihre Darstellung.* DIN 2331. 1980

[DG86] V. Duquenne, J.-L. Guigues: Famille minimale d'implication informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences Humaines* **24**(95), 1986, 5–18

[Du+01] V. Duquenne, C. Chabert, A. Cherfouh, J.–M. Delabar, A.–L. Doyen, D. Pickering: Structuration of phenotypes/genotypes through Galois lattices and implications. In: E. M. Nguifo, V. Duquenne, M. Liquiere (eds.): *Proc. ICCS-2001 Intl. Workshop on Concept Lattices–Based Theory, Methods, and Tools for Knowledge Discovery in Databases*, Stanford, July 2001, 21–34

[EGSW00] P. Eklund, B. Groh, G. Stumme, R. Wille: A Contextual-Logic Extension of TOSCANA. In: B. Ganter, G. W. Mineau (eds.): *Conceptual Structures: Logical, Linguistic, and Computational Issues.* Proc. ICCS '00. LNAI **1867**, Springer, Heidelberg 2000, 453–467

[FPSU96] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (eds.): *Advances in Knowledge Discovery and Data Mining.* AAAI/MIT Press, Cambridge 1996.

[Ga87] B. Ganter: Algorithmen zur Formalen Begriffsanalyse. In: B. Ganter, R. Wille, K. E. Wolff (eds.): *Beiträge zur Formalen Begriffsanalyse*, B. I.–Wissenschaftsverlag, Mannheim 1987, 241–254

[GK00] B. Ganter, S. O. Kuznetsov: Formalizing Hypotheses with Concepts. In: Ganter, B., Mineau, G. (Eds.): *Conceptual Structures: Logical, Linguistic and Computational Issues.* LNAI 1867.Springer, Berlin-Heidelberg-New York 2000, 342–356

[GSW86] B. Ganter, J. Stahl, R. Wille: Conceptual measurement and many-valued contexts. In: W. Gaul, M. Schader (eds.): *Classification as a tool of research.* North-Holland, Amsterdam 1986, 169–176

[GW99a] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations.* Springer, Heidelberg 1999. (Translation of: *Formale Begriffsanalyse: Mathematische Grundlagen.* Springer, Heidelberg 1996.)

[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

[Ge88] W. Geyer: *Lokale direkte Produkte von Begriffsverbänden.* Diplomarbeit, TH Darmstadt 1988

[GSW98] B. Groh, S. Strahringer, R. Wille: TOSCANA-systems based on thesauri. In: M.-L. Mugnier, M. Chein (eds.): *Conceptual Structures: Theory, Tools and Applications.* LNAI **1453**. Springer, Heidelberg 1998, 127–138.

[Gr94] T. Gruber: Towards principles for the design of ontologies used for knowledge sharing. *Intl. J. of Human and Computer Studies* **43**(5/6), 1994, 907–928

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

[HF95] J. Han, Y. Fu: Discovery of multiple-level association rules from large databases. *Proc. VLDB Conf.*, 1995, 420–431

[HPY00] J. Han, J. Pei, and Y. Yin: Mining frequent patterns without candidate generation. In *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, May 2000, 1–12

[HS02] S. Handschuh, S. Staab: Authoring and Annotation of Web Pages in CREAM. *Proc. World–Wide Web Conference (WWW 11)*, 2002, 462–473

[HSM01] S. Handschuh, S. Staab, A. Mädche: CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. *Proc. 1st Intl. Conf. on Knowledge Capture.* ACM Press, New York 2001, 76–83

[Hd01] J. Hendler: Agents and the Semantic Web. *IEEE Intelligent Systems* **16**(2), 2001, 30–37

[Hn74] H. von Hentig: *Magier oder Magister? Über die Einheit der Wissenschaft im Verständigungsprozess.* 1. Aufl., Suhrkamp, Frankfurt 1974

[HS01] J. Hereth, G. Stumme: Reverse Pivoting in Conceptual Information Systems. In: H. Delugach, G. Stumme (Eds.): *Conceptual Structures: Broadening the Base.* Proc. ICCS '01. LNAI **2120**, Springer, Heidelberg 2001, 202–215

[HSWW00] J. Hereth, G. Stumme, R. Wille, U. Wille: Conceptual Knowledge Discovery and Data Analysis. In: B. Ganter, G. W. Mineau (eds.): *Conceptual Structures: Logical, Linguistic, and Computational Issues.* Proc. ICCS '00. LNAI **1867**, Springer, Heidelberg 2000, 421–437

[HMWG98] J. Hipp, A. Myka, R. Wirth, U. Güntzer: A new algorithm for faster mining of generalized association rules. In: Jan M. Zytkow, Mohamed Quafalou (eds.): *Principles of Data Mining and Knowledge Discovery.* Proc. PKDD '98. LNAI **1510**, Springer, Heidelberg 1998, 74–82

[Ho95] D. Horster: Habermas, Jürgen. In: B. Lutz (ed.): *Metzler Philosophen Lexikon. Von den Vorsokratikern bis zu den Neuen Philosophen.* Metzler, Stuttgart–Weimar 1995, 335–341

[Ho98] E. Hovy: Combining and standardizing large-scale, practical ontologies for machine translation and other uses. *Proc. 1st Intl. Conf. on Language Resources and Evaluation*, Granada, Spain, May 1998

[ISO704] International Organization of Standardization: *ISO 704. Terminology Work — Principles and Methods.* 2000

[JKN01] M. Jarke, R. Klemke, A. Nick: Broker's lounge — an environment for multidimensional user-adaptive knowledge management. *Proc. 34th Hawaii Intl. Conf. on System Siences (HICSS-34)*, 2001, 83

[Jo99] K. Jones: *View Mail Users Manual. http://www.wonderworks.com/vm.* 1999

[KHC97] M. Kamber, J. Han, and Y. Chiang. Metarule-guided mining of multidimensional association rules using data cubes. *Proc. of the 3rd KDD Int'l Conf.*, Aug. 1997. 207–210

[KN95] R. E. Kent, C. Neuss: Creating a Web Analysis and Visualization Environment. *Computer Networks and ISDN Systems* **28**(1&2), 1995, 109–117

[Kl89] A. M. Kleinhans: *Wissensverarbeitung im Management. Möglichkeiten und Grenzen wissensbasierter Managementunterstützungs-, Planungs- und Simulationssysteme.* PhD thesis, Univ. Stuttgart. Verlag Peter Lang, Frankfurt 1989

[KMRTV94] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, A. I. Verkamo: Finding interesting rules from large sets of discovered association rules. *Proc. CIKM Conf.*, 1994, 401–407

[KM88] U. Klotz, A. Mann: *Begriffexploration.* Diplomarbeit, TH Darmstadt 1988

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

[Ko89] W. Kollewe: Evaluation of a survey with methods of formal concept analysis. In: O.Opitz (ed.): *Conceptual and numerical analysis of data.* Springer-Verlag, Berlin-Heidelberg 1989, 123–134

[LeW95] F. Lehmann, R. Wille: A triadic approach to formal concept analysis. In: G. Ellis, R. Levinson, W. Rich, J. F. Sowa (eds.): *Conceptual structures: applications, implementation and theory.* LNAI **954**. Springer, Berlin–Heidelberg–New York 1995, 32–43

[LSW97] B. Lent, A. Swami, and J. Widom: Clustering association rules. In *Proc. of the 13th Int'l Conf. on Data Engineering (ICDE)*, March 1997, 220–231

[LK98] D. Lin, Z. M. Kedem: Pincer-Search: A new algorithm for discovering the maximum frequent sets. *Proc. of the 6th Int'l Conf.on Extending Database Technology (EDBT)*, Mar. 1998, 105–119

[LuW91] P. Luksch, R. Wille: A mathematical model for conceptual knowledge systems. In: H.-H. Bock, P. Ihm (eds.): *Classification, data analysis, and knowledge organization.* Springer, Heidelberg 1991, 156–162

[Lu91] M. Luxenburger: Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines*, 29(113), 1991, 35–55

[Lu93] M. Luxenburger: Partial implications. Part I of *Implikationen, Abhängigkeiten und Galois Abbildungen.* PhD thesis, TU Darmstadt. Shaker, Aachen 1993

[Mä02] A. Mädche: *Ontology Learning for the Semantic Web.* PhD thesis, Universität Karlsruhe. Kluwer, Dordrecht 2002

[Ma97] H. Mannila: Methods and problems in data mining. *Proc. of the 6th Int'l Conf. on Database Theory (ICDT)*, Jan. 1997, 41–45

[MT97] H. Mannila, H. Toivonen: Levelwise Search and Borders of Theories in Knowledge Discovery. *Data Mining and Knowledge Discovery* **1**(3), 1997, 241–258

[MFRW00] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder: An environment for merging and testing large Ontologies. *Proc. 7th Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'2000)*, Breckenridge, Colorado, USA, April 2000, 483–493

[MPC96] R. Meo, G. Psaila, S. Ceri: A new SQL-like operator for mining association rules. *Proc. VLDB Conf.*, 1996, 122–133

[Mi80] R. S. Michalski: Knowledge acquisition through conceptual clustering: a theoretical framework and an algorithm for partitioning data into conjunctive concepts. *Policy Analysis and Information Systems* **4**(3), 1980, 219–244

[MG95] G. Mineau, R. Godin: Automatic Structuring of Knowledge Bases by Conceptual Clustering. *IEEE Transactions on Knowledge and Data Engineering* **7**(5),1995, 824–829

[MSW99] G. Mineau, G. Stumme, R. Wille: Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis. In: W. Tepfenhart, W. Cyre (eds.): *Conceptual Structures: Standards and Practices.* Proc. ICCS '99. LNAI **1640**. Springer, Heidelberg 1999, 423–441

[Mo93] K. Morik: Balanced corporative modeling. *Machine Learning* **11**, 217–235

[NLP98] R. T. Ng, V. S. Lakshmanan, J. Han, A. Pang: Exploratory mining and pruning optimizations of constrained association rules. *Proc. SIGMOD Conf.*, 1998, 13–24

[NM00] N. Fridman Noy, M. A. Musen: PROMPT: algorithm and tool for automated ontology merging and alignment. *Proc. 17th Natl. Conf. on Artificial Intelligence (AAAI'2000)*, Austin, TX, July/August 2000, 450–455

[PCY95] J. S. Park, M. S. Chen, P. S. Yu: An efficient hash based algorithm for mining association rules. *Proc. ACM SIGMOD Int'l Conf. on Management of Data*, May 1995, 175–186

[PBTL98] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Pruning closed itemset lattices for association rules. *Actes des 14ièmes journées Bases de données avancées*, Oct. 1998, 177–196

[PBTL99a] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Efficient mining of association rules using closed itemset lattices. *Journal of Information Systems* **24**(1), 1999, 25–46

[PBTL99b] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal: Discovering frequent closed itemsets for association rules. *Proc. of the 7th Int'l Conf. on Database Theory (ICDT)*, Jan. 1999, 398–416

[PHM00] J. Pei, J. Han, R. Mao: Closet: An efficient algorithm for mining frequent closed itemsets. *Proc. Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD)*, May 2000, 21–30

[Pe31] Ch. S. Peirce: *Collected Papers.* Harvard University Press, Cambridge 1931–35

[Po97] S. Pollandt: *Fuzzy Begriffe: Formale Begriffsanalyse von unscharfen Daten.* Springer, Berlin-Heidelberg 1997

[Pr97] S. Prediger: Logical scaling in 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, 332–341.

[Pr98] S. Prediger: *Kontextuelle Urteilslogik mit Begriffsgraphen. Ein Beitrag zur Restrukturierung der mathematischen Logik.* Dissertation, TU Darmstadt. Shaker Verlag, Aachen 1998

[Pr00] S. Prediger: Mathematische Logik in der Wissensverarbeitung. Historisch–philosophische Gründe für eine Kontextuelle Logik. *Math. Semesterberichte* **47**(2), 2000, 165–191

[PS99] S. Prediger, G. Stumme: Theory-Driven Logical Scaling. In: E. Franconi et al (eds.): *Proc. 6th Intl. Workshop Knowledge Representation Meets Databases.* CEUR Workshop Proc. **21**, 1999. Also in: P. Lambrix et al (eds.): *Proc. Intl. Workshop on Description Logics (DL '99).* CEUR Workshop Proc. **22**, 1999 (http://CEUR-WS.org/Vol-21)

[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

[PRR99] G. Probst, S. Raub, K. Romhardt: *Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen.* Th. Gabler Verlag, Wiesbaden 1999

[RC97] D. Richards, P. Compton: Combining Formal Concept Analysis and Ripple Down Rules to Support Reuse. *Proc. 9th Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE '97)*, Springer 1997

[RW00] T. Rock, R. Wille: Ein TOSCANA-System zur Literatursuche. In: G. Stumme and R. Wille (eds.): *Begriffliche Wissensverarbeitung: Methoden und Anwendungen.* Springer, Berlin-Heidelberg 2000, 239–253

[SON95] A. Savasere, E. Omiecinski, S. Navathe: An efficient algorithm for mining association rules in large databases. *Proc. of the 21th Int'l Conf. on Very Large Data Bases (VLDB)*, Morgan Kaufmann, San Franisco 1995, 432–444

[SZ92] G. Schmidt, M. Zickwolff: Cases, models and integrated knowledge acquisition to formalize operators in manufacturing. *Proc. Knowledge acquisition for knowledge-based systems workshop.* Banff, Canada 1992

[SS98] I. Schmitt, G. Saake: Merging inheritance hierarchies for database integration. *Proc. 3rd IFCIS Intl. Conf. on Cooperative Information Systems*, New York City, Nework, USA, August 20-22, 1998, 122–131

[Sc+00] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. R. Shadbolt, W. Van de Velde, B. Wielinga: *Knowledge Engineering and Management.* MIT Press 2000

[Sc90] E. Schröder: *Algebra der Logik I, II, III.* 1890, 1891, 1895. Thoemmes Press, Bristol 2001

[SBM98] C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, **2**(1), 1998, 39–68

[Sn96] G. Snelting: Reengineering of Configurations Based on Mathematical Concept Analysis. *ACM Transactions on Software Engineering and Methodology* **5**(2), 1996, 146–189

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

[SpW89] N. Spangenberg, K. E. Wolff: Comparison between principal component analysis and formal concept analysis of repertory grids. In: W. Lex (ed.): *Arbeitstagung Begriffsanalyse und Künstliche Intelligenz*, Informatik–Bericht 89/3. TU Clausthal, 1991, 127–134

[SA95] R. Srikant, R. Agrawal: Mining generalized association rules. In: U. Dayal, P. M. D. Gray, S. Nishio (eds.): *Proc. VLDB Conf.*, Morgan–Kaufman, San Francisco 1995, 407–419

[SVA97] R. Srikant, Q. Vu, R. Agrawal: Mining association rules with item constraints. D. Heckerman, H. Mannila, D. Pregibon, R. Uthurusamy (eds.): *Proc. KDD Conf.*, AAAI Press, Menlo Park 1997, 67–73

[Sta+00] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Mädche, R. Studer, Y. Sure: Semantic Community Web Portals. *Proc. 9th World Wide Web Conference (WWW 9).* Amsterdam 2000, 473–491

[StaM01] S. Staab, A. Mädche: Knowledge Portals — Ontologies at Work. *AI Magazine* **21**(2), 2001

[SSSS01] S. Staab, H.–P. Schnurr, R. Studer, Y. Sure: Knowledge Processes and Ontologies. *IEEE Intelligent Systems* **16**(1), 2001

[SMSSS01] N. Stojanovic, A. Mädche, S. Staab, R. Studer, Y. Sure: SEAL — A Framework for Developing SEmantic portALs. In: *Proc. 1st Intl. Conf. on Knowledge Capture (K–CAP '01).* ACM Press, New York 2001, 155–162

[StrW93] S. Strahringer, R. Wille: Conceptual clustering via convex-ordinal structures. In: O. Opitz, B. Lausen, R. Klar (eds.): *Information and Classification.* Springer, Berlin-Heidelberg 1993, 85–98

[SDFS00] R. Studer, S. Decker, D. Fensel, S. Staab: Situation and Perspective of Knowledge Engineering. In: J.Cuena, Y.Demazeau, A. Garcia, J.Treur (eds.): *Knowledge Engineering and Agent Technology.* IOS Series on Frontiers in Artificial Intelligence and Applications. IOS Press, 2000

[St95] G. Stumme: Knowledge Acquisition by Distributive Concept Exploration. In: G. Ellis, R. A. Levinson, W. Rich, J. F. Sowa (eds.): *Suppl. Proc. of the Third International Conference on Conceptual Structures*, Santa Cruz, CA, USA, August 1995, 98–111

[St96a] G. Stumme: Local Scaling in Conceptual Data Systems. In: P. W. Eklund, G. Ellis, G. Mann (eds.): *Conceptual Structures: Knowledge Representation as Interlingua.* Proc. ICCS '96. LNAI **1115**, Springer, Heidelberg 1996, 308–320

[St96b] G. Stumme: The Concept Classification of a Terminology Extended by Conjunction and Disjunction. In: N. Foo, R. Goebel (eds.): *PRICAI '96: Topics in Artificial Intelligence.* Proc. PRICAI '96. LNAI **1114**, Springer, Heidelberg 1996, 121–131

[St96c] G. Stumme: Exploration tools in Formal Concept Analysis. In: *Ordinal and Symbolic Data Analysis.* Studies in classification, data analysis, and knowledge organization **8**, Springer, Heidelberg 1996, 31–44

[St97] G. Stumme: *Concept Exploration — Knowledge Discovery in Conceptual Knowledge Systems.* PhD thesis, TU Darmstadt. Shaker, Aachen 1997

[St98] G. Stumme: Exploring Conceptual Similarities of Objects for Analyzing Inconsistencies in Relational Databases. *Proc. Workshop on Knowledge Discovery and Data Mining, 5th Pacific Rim Intl. Conf. on Artificial Intelligence*. Singapore, Nov. 1998, 41–50

[St99a] G. Stumme: Hierarchies of Conceptual Scales. In: B. Gaines, R. Kremer, M. Musen (eds.): *Proc. Workshop on Knowledge Acquisition, Modeling and Management*, Vol. 2. Banff, Oct. 1999, pages 5.5.1–18

[St99b] G. Stumme: *Conceptual Knowledge Discovery with Frequent Concept Lattices.* FB4-Preprint **2043**, TU Darmstadt 1999

[St99c] G. Stumme: Dual Retrieval in Conceptual Information Systems. In: A. Buchmann (ed.): *Datenbanksysteme in Büro, Technik und Wissenschaft.* Proc. BTW '99. Springer, Heidelberg 1999, 328–342

[St00] G. Stumme: Conceptual On-Line Analytical Processing. In: K. Tanaka, S. Ghandeharizadeh, Y. Kambayashi (eds.): *Information Organization and Databases.* Chpt. 14. Kluwer, Boston–Dordrecht–London 2000, 191–203

[St02a] G. Stumme: *Conceptual Knowledge Discovery and Processing.* Habilitation thesis, Universität Karlsruhe 2002

[St02b] G. Stumme: Formal Concept Analysis on its Way from Mathematics to Computer Science. Invited Talk. In: U. Priss, D. Corbett, G. Angelova (Eds.): *Conceptual Structures: Integration and Interfaces*, Proc. ICCS 2002, LNAI **2393**, Springer, Heidelberg 2002, 2–19

[SHB01] G. Stumme, A. Hotho, B. Berendt (eds.): *Semantic Web Mining.* Proc. of the Semantic Web Mining Workshop of the 12th Europ. Conf. on Machine Learning (ECML'01) / 5th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'01), Freiburg, September 3rd, 2001

[SM01] G. Stumme, A. Mädche: FCA–Merge: Bottom-Up Merging of Ontologies. *Proc. 17th Intl. Conf. on Artificial Intelligence (IJCAI '01).* Seattle, WA, USA, 2001, 225–230

[STBPL00] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal: Fast Computation of Concept Lattices Using Data Mining Techniques. *Proc. 7th Intl. Workshop on Knowledge Representation Meets Databases*, Berlin, 21–22. August 2000. CEUR-Workshop Proc. http://CEUR-WS.org/Vol-29

[STBPL01a] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, L. Lakhal: Intelligent Structuring and Reducing of Association Rules with Formal Concept Analysis. In: F. Baader. G. Brewker, T. Eiter (eds.): *KI 2001: Advances in Artificial Intelligence.* Proc. KI 2001. LNAI **2174**, Springer, Heidelberg 2001, 335–350

[STBPL02] G. Stumme, R. Taouil, Y. Bastide, N. Pasqier, L. Lakhal: Computing Iceberg Concept Lattices with Titanic. *J. on Knowledge and Data Engineering* **42**(2), 2002, 189–222

[SWW98] G. Stumme, R. Wille, U. Wille: Conceptual Knowledge Discovery in Databases Using Formal Concept Analysis Methods. In: J. M. Żytkow, M. Quafofou (eds.): *Principles of Data Mining and Knowledge Discovery.* Proc. PKDD '98, LNAI **1510**, Springer, Heidelberg 1998, 450–458

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

[SW97] G. Stumme, K. E. Wolff: Computing in Conceptual Data systems with relational structures. *Proc. Intl. Conf. on Knowledge Retrieval, Use, and Storage for Efficiency*, Vancouver, Canada, 11.–13. 8. 1997, 206–219

[SW98] G. Stumme, K. E. Wolff: Numerical Aspects in the Data Model of Conceptual Information Systems. In: Y. Kambayashi, Dik Kun Lee, Ee-Peng Lim, M. K. Mohania, Y. Masunaga (eds.): *Advances in Database Technologies. Proc. Intl. Workshop* *on Data Warehousing and Data Mining, 17th Intl. Conf. on Conceptual Modeling (ER '98).* LNCS **1552**, Springer, Heidelberg 1999, 117–128

[SMS00] Y. Sure, A. Mädche, S. Staab: Leveraging corporate skill knowledge — from ProPer to OntoProper. In: D. Mahling. U. Reimer (eds.): *Proc. 3rd Intl. Conf. on Practical Aspects of Knowledge Management*, Basel, October 2000. www.research.swisslife.ch/pakm2000

[To01] *The ToscanaJ-Project: An Open-Source Reimplementation of Toscana.* http://toscanaj.sourceforge.net

[To96] H. Toivonen: Sampling large databases for association rules. *Proc. of the 22nd Int'l Conf. on Very Large Data Bases (VLDB)*, Sept. 1996, 134–145

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

[Wd95] M. Wild: Computations with finite closure systems and implications. In: D.–Z. Du, M. Li (eds.): *Computing and combinatorics.* LNCS 959. Springer, Berlin-Heidelberg 1995, 111–120

[Wi82] R. Wille: Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets.* Reidel, Dordrecht–Boston, 445–470

[Wi83] R. Wille: Subdirect decomposition of concept lattices. *Algebra Universalis* **17**, 1983, 275-287

[Wi84] R. Wille: Line diagrams of hierarchical concept systems. *Int. Classif.* **11**, 77–86. Translation of: Liniendiagramme hierarchischer Begriffssysteme. In: H.-H. Bock (ed.): *Anwendungen der Klassifikation: Datenanalyse und numerische Klassifikation.* Indeks-Verlag, Frankfurt, 1984, 32–51

[Wi87] R. Wille: Bedeutungen von Begriffsverbänden. In: B. Ganter, R. Wille, K. E. Wolff (eds.): *Beiträge zur Begriffsanalyse.* B.I.–Wissenschaftsverlag, Mannheim 1987, 161–211

[Wi88] R. Wille: Allgemeine Wissenschaft als Wissenschaft für die Allgemeinheit. In H. Böhme, H.–J. Gamm (eds.): *Verantwortung in der Wissenschaft.* TH Darmstadt 1988. Reprinted in: *Conceptus — Zeitschrift für Philosphie* **60**, 1989, 117–128

[Wi94] R. Wille: Plädoyer für eine philosophische Grundlegung der Begrifflichen Wissensverarbeitung. In: R. Wille, M. Zickwolff (eds.): *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 und P. Agliano (eds.): *Logic and Algebra.* Marcel Dekker, New York 1996, 267-281

[Wi97] 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.* Proc. ICCS '97. LNAI **1257**. Springer, Heidelberg 1997, 290–303

[Wi99] R. Wille: Conceptual landscapes of knowledge: a pragmatic paradigm for knowledge processing. In: W. Gaul, H. Locarek-Junge (eds.): *Classification in the Information Age.* Springer, Heidelberg 1999, 344–356

[Wi00] R. Wille: Contextual logic summary. In: G. Stumme (ed.): *Working with Conceptual Structures.* Suppl. Proc. ICCS 2000. Shaker, Aachen 2000, 265–276

[Wi01a] R. Wille: Begriffliche Wissensverarbeitung: Theorie und Praxis. *Informatik Spektrum* **23**(6), 2000, 357–369

[Wi01b] R. Wille: Why can concept lattices support knowledge discovery in databases? In: E. M. Nguifo, V. Duquenne, M. Liquiere (eds.): *Concept Lattice-based theory, methods and tools for Knowledge Discovery in Databases.* Proc. of Workshop of the

9th Intl. Conf. on Conceptual Structures (ICCS '01). http://CEUR-WS.org/Vol-42/
[WZ94] R. Wille, M. Zickwolff (eds.): *Begriffliche Wissensverarbeitung — Grundfragen und Aufgaben.* B. I.–Wissenschaftsverlag, Mannheim 1994
[WMJ00] S. Wrobel, K. Morik, T. Joachims: Maschinelles Lernen und Data Mining. In: G. Görz, C.–R. Rollinger, J. Schneeberger (eds.): *Handbuch der Künstlichen Intelligenz.* 3. Auflage. Oldenbourg, München–Wien 2000, 517–597
[ZPOL97] M. J. Zaki, S. Parthasarathy, M. Ogihara, W. Li: New algorithms for fast discovery of association rules. D. Heckerman, H. Mannila, D. Pregibon (eds.): *Proc. 3rd Intl. Conf. on Knowledge Discovery and Data Mining (KDD-97)*, Newport Beach, California, USA, August 14-17, 1997. AAAI Press 1997, 283–286
[Za00] M. J. Zaki: Generating non-redundant association rules. *Proc. 6th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, August 20-23, 2000, Boston, MA, USA. ACM, 2000. 34–43
[Zi92] M. Zickwolff: *Begriffliche Wissenssysteme in der Künstlichen Intelligenz.* FB4-Preprint 1506, TH Darmstadt 1992