Towards an Order-Theoretical Foundation for Maintaining and Merging Ontologies*

Gerd Stumme, Rudi Studer, York Sure

Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Universität Karlsruhe (TH), D–76128 Karlsruhe, {stumme,studer,sure}@aifb.uni-karlsruhe.de

Abstract

This paper discusses steps towards an order-theoretic foundation for maintaining and merging ontologies. We claim that the commitment to ontologies having certain structural properties allows a semantically founded (semi-)automation of operations on ontologies. We provide a list of research questions and give first answers. In particular we argue for the use of Formal Concept Analysis for maintaining ontologies.

1 Introduction

Business processes are typically treated in a cooperation of several departments of a company or even across company boundaries. This requires the management of knowledge related to the business processes across department and company boundaries. Ontologies have turned out to be a successful approach for structuring such informal, semi-formal, and formal knowledge.

Nevertheless it is impossible in practice to provide a single, company-wide ontology satisfying all users with regard to coverage, precision, actuality, and individualization. Different departments need specific approaches and vocabularies for describing and solving their specific tasks. Hence the balance between the two conflicting objectives of providing a common knowledge core on the one hand and sufficient influence of the different departments on its structure and content on the other hand has to be maintained.

The approach presented in this paper attacks that problem by providing multiple ontologies which are organized along the organizational structure of the

*This research is partially supported as project STU 170/11–1 by the Deutsche Forschungsgemeinschaft (DFG).

company (departments, working groups, etc.) and/or the structure of the business processes. There may be a general, company-wide ontology which specifies the common vocabulary. Subdivisions will then have their own, more specific ontologies with different levels of granularity. The smaller a unit is, the more (task-)specific will be its ontology.

When a business process passes from one unit to another, the need for communication arises. At that moment, the underlying ontologies have to be made compatible. Compatibility can be obtained by *merging* the ontologies into a unique one, or by *aligning* them. Merging two ontologies means creating a new ontology in a semi-automatic manner by using the concepts of both ontologies and by identifying some of them. Aligning two ontologies means defining a mapping between the two ontologies which translates concepts of the first ontology into the second one.

There is quite some research on the problems which come along with merging and aligning ontologies. References are given in Section 2. Many of them present tools which support the interactive process of merging/aligning. The presentation is mostly done from a software-engineering point of view, rather than from a structure-theoretic point of view. Our approach is based on the latter, with the aim to provide formal semantics for the operations which are independent from the actual implementation.

Our first attempts to provide such a formalization showed that at this technical level some sensitive decisions have to be made. The exact definition of what an ontology is, and the specification of the constraints influences the limitation of the power of the merging mechanism on the one hand, and allows to derive structural results about the target ontology on the other hand. These structural results can be used to infer properties which the target ontology inherits from the source ontologies.

In this paper, we present our first steps towards an order-theoretical foundation for maintaining and merging ontologies. As the research project is at its very beginning, we present our approach, and the research program to be followed in the next future.

2 Related Work

A first approach for supporting the merging of ontologies is described in [Ho98]. There, several heuristics are described for identifying corresponding concepts in different ontologies, e.g. the NAME heuristic compares the names of two concepts, the DEFINITION heuristic uses linguistic techniques for comparing the natural language definitions of two concepts, or the TAXONOMY heuristic checks the closeness of two concepts to each other. Whereas such heuristics may be used as building blocks in an overall framework, the described approach does not support a task- and thus goal-oriented integration of ontologies.

The OntoMorph system [Ch00] offers two kinds of mechanisms for translating and merging ontologies: syntactic rewriting supports the translation between two different knowledge representation languages, semantic rewriting offers means for inference-based transformations that exploit the PowerLoom knowledge representation system. OntoMorph puts emphasis on the transformation of ontologies into a common format. This may be seen as one step in the merging process of ontologies. OntoMorph explicitly allows to violate the preservation of semantics in trade-off for a more expressive, flexible transformation mechanism.

In [Mc+00] the Chimaera system is described that provides support for merging of ontological terms from different sources, for checking the coverage and correctness of ontologies and for maintaining ontologies over time. Chimaera handles OKBC [C+98] compliant ontologies and supports the merging of ontologies (i) by coalescing two semantically identical terms from different ontologies and (ii) by identifying terms that should be related by subsumption or disjointness relationships. Chimaera offers a broad collection of commands and functions to support the merging and diagnostic process. However, the underlying assumptions about structural properties of the ontologies at hand are not made explicit.

PROMPT [NM00] is an algorithm for ontology merging and alignment that is able to handle ontologies that are specified in a OKBC compatible format [C+98]. PROMPT starts with the identification of matching class names. Based on this initial step an iterative approach is carried out for performing automatic updates, finding resulting conflicts, and making suggestions to remove these conflicts. PROMPT is implemented as an extension to the Protégé-2000 knowledge acquisition tool and offers a collection of operations for merging two classes and related slots. However, the description of PROMPT in [NM00] does not specify the structural constraints that have to be met by the source ontologies as well as by the target ontology resulting from the merging process.

3 Formalization of the Problem

The references given in the last section provide techniques for maintaining, merging, and aligning ontologies. However, most of these approaches focus more on the software engineering aspects involved in these tasks, rather than on structural aspects of the ontologies. In this paper, we outline our approach which is guided by ideas from order and lattice theory. We believe that structural properties of ontologies can be used to support their maintenance. In this section, we outline the problems which shall be investigated in the light of order and lattice theory.

3.1 Maintaining Ontologies

We discuss two questions: the first deals with the definition of what an ontology is, and the second with various topics for the maintenance of ontologies.

3.1.1 What do we consider as an ontology?

There are different 'definitions' in the literature of what an ontology should be. Some of them are discussed in [Gu97], the most prominent being "An ontology is an explicit specification of a conceptualization" [Gr94]. Common to all these definitions is their level of generalization. The reason is that the definition should cover all different kinds of ontologies, and should not be related to a particular method of knowledge representation [HSW97]. However, as we want to study structural aspects, we have to commit ourselves to one specific ontology representation framework, and to a precise, detailed definition. In the sequel of this paper, we will consider only the common core of all ontologies, namely the IS_A relation. We plan to extend this approach to ontologies based on F-Logic [KLW95] as used for instance in Ontobroker [D+99] and OntoEdit [SM00].

3.1.2 How to support the maintenance of ontologies?

It is a common understanding that in most applications the maintenance of an ontology has to involve a human expert who is responsible for the structure and the content of the ontology. Since the maintenance is tedious and time-consuming, and the domain expert risks to loose orientation in large ontologies, tool support is needed which makes reasonable suggestions to the expert or automates certain tasks based on principles the expert confirmed beforehand. In this paper we present an approach based on Formal Concept Analysis [GW99] which suggests new concepts to the domain expert. Future research shall include the integration of approaches for generating ontologies from text with linguistic methods, and for evaluating them with data mining techniques [MS00a, MS00b].

3.2 Merging Ontologies

Merging ontologies mean producing a target ontology out of two (or more) source ontologies. We discuss some questions about how a structural approach can improve the merging process.

3.2.1 Which consistency conditions should ontologies verify in order to be allowed to be merged?

Agreeing on a number of fixed consistency conditions allows the user a better understanding of the relationship between the source ontologies and the target one. For instance, if one requires that each concept in a source ontology has a counterpart in the target ontology, then the user knows that he can formulate all his queries in the target ontology only. On the other hand, if there are conflicting entries in the two ontologies, then it may not be possible to include all concepts. Other questions are: Shall it be allowed to assign different concepts of one source ontology to the same concept in the target ontology? And, vice versa, shall it be possible that there is a concept in a source ontology which is represented by more than one concept (or none at all) in the target ontology?

3.2.2 Can the merging of ontologies be described as a (parametrized) operation on the set of ontologies?

The aim is to reach a structural description of the merging process by providing a denotational semantic instead of a purely operational one. While the latter one is implementation-dependent, the former allows to deduce general properties of the relationship between source and target ontologies. The research on this topic will include a comparison and eventually an integration with the Scalable Knowledge Composition project of Stanford University ([Mi+00], see also http://www-db.stanford.edu/SKC).

3.2.3 Which properties does the target ontology have in function of the source ontologies?

Using the description obtained in the last paragraph, one can analyze which properties of the source ontologies are inherited by the target one. This information allows to provide a consistent dialogue with the user based on explicitly stated assumptions. In particular one can choose appropriate knowledge visualization tools which rely on these assumptions.

3.2.4 How can the merging operation be embedded in an interactive method for determining its parameters?

Merging ontologies by hand is a time-consuming process. The tools presented in Section 2 provide heuristics for making suggestions to the user. We anticipate that structural knowledge about the merging operation can enhance these heuristics.

3.2.5 How can the definition of the operation be used for a (semi-)automatic update of the target ontology when the source ontologies change?

Enterprises constantly adapt themselves to new business conditions. This implies that the ontologies of their units undergo constant change as well. The target ontology also has to be adapted in function of these changes. Because the adaptation is a permanent process, it is desirable to automate it as much as possible. The stronger the information about relationships between the source and the target ontologies is, the easier will it be to realize the automation.

3.2.6 How can other relations beside the is_a relation be integrated?

We start our research project with the IS_A relation in order to benefit from the results of many years of research in order theory. Nevertheless ontologies consist of more than this relation, with decreasing amount of structural properties. The PART_OF relation for instance is a relatively common relation in ontologies. Although treated as a partial order in many implementations, there are numerous examples showing that the assumption of its transitivity is not always justified. The fewer information is given about the specific relations, the more one has to fall back on heuristics. The interplay between structural information and the heuristics has to be studied.

3.2.7 How can meta-knowledge about concepts and relations provided by axioms be exploited for the merging process?

Relationships between different kinds of relations can be expressed in logical axioms. In a skill management application, for instance, such an axiom can state that a person who worked in a chemistry project has to be in the HAS_EXPERTISE_IN relation to the concept CHEMISTRY. Using this information in an inference engine like OntoBroker [D+99], one can overcome some of the drawbacks of relationships with few structural properties.

3.3 Aligning Ontologies

Aligning two ontologies means constructing a function between the ontologies which assigns concepts of the first ontology to 'similar' concepts in the second one. The following questions arise from formally analyzing the alignment scenario.

3.3.1 Which properties should a function have which describes the alignment between two ontologies?

The alignment of two ontologies is considered as a mapping from the concepts (and the relations) of one ontology to the concepts (relations) of the second ontology. Should this mapping necessarily be a completely defined function? Or may it be partially defined only, when the concepts of the first ontology do not have a counterpart in the second one? Should the mapping preserve the order of the IS_A relation? Should it be injective, or should one allow to map two different concepts of the first ontology on the same concept of the second? How do these decisions influence recall, precision, and understandability of the result of a retrieval process using this alignment function?

3.3.2 How can an interactive knowledge acquisition process support the construction of the aligning function?

Depending on the properties required from the alignment function, an interactive tool can make suggestions to the human expert which are coherent with the properties. This means in particular that when source and target ontology share some property, it is a reasonable assumption for the aligning function to preserve that property. Systems like Chimaera or PROMPT can then reduce their suggestions to those functions.

3.3.3 How can meta-knowledge about concepts and relations provided by axioms be exploited for the aligning process?

As in the merging process, one can benefit from inferences derived from logical axioms. These decrease the number of possible alignment functions to be considered.

4 Maintaining Ontologies by Applying Formal Concept Analysis

Since concepts are necessary for expressing human knowledge the knowledge management process benefits from a comprehensive formalization of concepts. Formal Concept Analysis [Wi82, GW99] offers such a formalization by mathematizing the concept 'concepts' as a unit of thought constituted of two parts: its extension and its intension. This understanding of 'concept' is first mentioned explicitly in the Logic of Port Royal [AN68] and has been established in the German standards DIN 2330 and DIN 2331. Here we apply this understanding of concepts to the concepts given in ontologies.

4.1 Formal Concept Analysis

We recall the basics of Formal Concept Analysis as far as they are needed for this paper. A more extensive overview is given in [GW99]. To allow a mathematical description of extensions and intensions, Formal Concept Analysis starts with a *(formal) context* defined as a triple $\mathbb{K} := (G, M, I)$, where G is a set of *objects*, M is a set of *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".

Example. Figure 1 shows a formal context about dependencies of throughput times in production processes (refer to [Sch97, p. 238]). The objects are seven types of throughput times, and the attributes are four influence factors for the duration of the types. The binary relation (i. e., the cross-table) indicates which type is influenced by which factor.

For $A \subseteq G$, we define $A' := \{m \in M \mid \forall g \in A\colon (g, m) \in I\}$ and, for $B \subseteq M$, we define $B' := \{g \in G \mid \forall m \in B\colon (g, m) \in I\}$.

A *formal concept* of a formal context (G, M, I) is defined as 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). 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

	Losgröße	Arbeitsgang	Arbeitsgangfolge	Betriebsmittelgruppe
Rüstzeit		×	×	×
Bearbeitungszeit	×			×
Nachbearbeitungszeit	×	×		×
Wartezeit bis Transport		×		
Transport	×		×	
Wartezeit in Auftragswarteschlange				×
Vorbereitungszeit	×	×		×

Figure 1: A formal context about dependencies of time components of orders

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.

In the *line diagram*, the name of an object g is always attached to the circle representing the smallest concept with g in its extent; dually, the name of an attribute m is always attached to the circle representing the largest concept with m in its intent. This allows us to read the context relation from the diagram because an object g has an attribute m if and only if there is an ascending path from the circle labeled by g to the circle labeled by m. The extent of a concept consists of all objects whose labels are below in the diagram, and the intent consists of all attributes attached to concepts above in the hierarchy.

Example. In the concept lattice in Figure 2, the concept in the middle without any label has {Vorbereitungszeit, Nacharbeitungszeit, Rüstzeit} as extent, and {Betriebsmittelgruppe, Arbeitsgang} as intent.

The concept lattice provides a conceptual clustering to the user. Each concept extent is a cluster which is described by its intent. The line diagram shows the hierarchical structure of the clusters. Furthermore it can be used to examine dependencies (implications) and independencies between the attributes: For $X, Y \subseteq M$, we say that the *implication* $X \to Y$ *holds* in the context, if each object having all attributes in X also has all attributes in Y.

Example. The implication {Losgröße, Arbeitsgang} $\to$ {Betriebsmittelgruppe} holds in the context. It can also be read directly in the line diagram: the largest concept having both 'Losgröße' and 'Arbeitsgang' in its intent is the one labeled by 'Vorbereitungszeit' and 'Nachbereitungszeit', and it has additionally 'Betriebsmittelgruppe' in its intent. Hence all throughput times influenced by 'Losgröße'

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

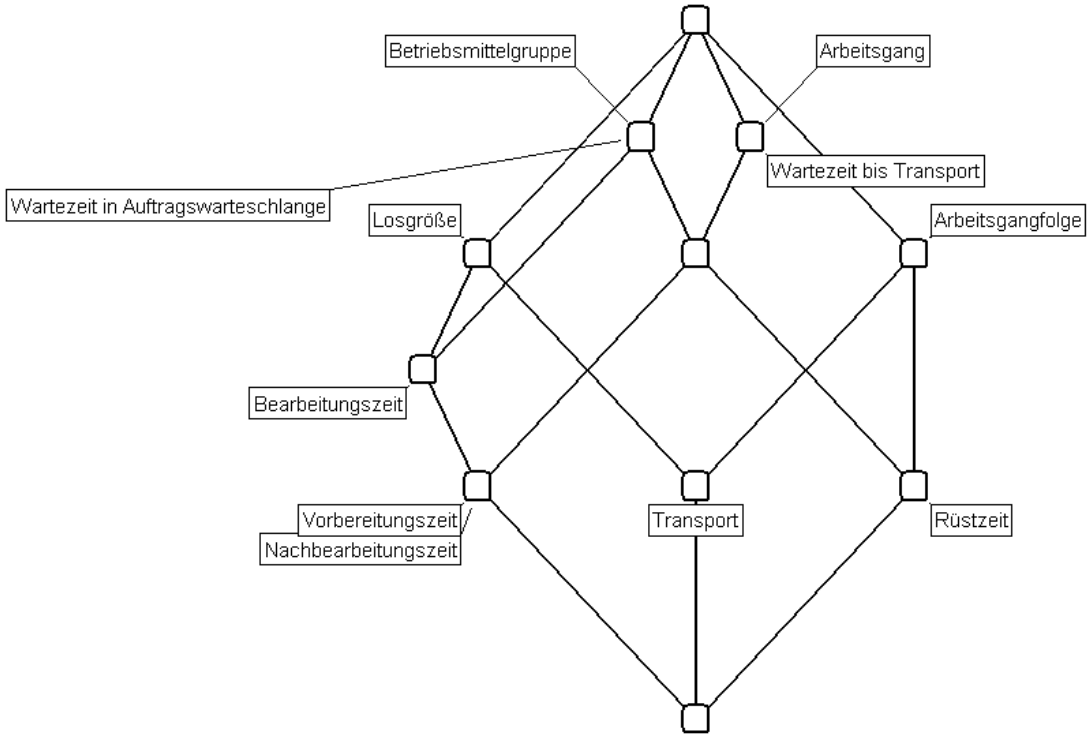

Figure 2: The concept lattice of the formal context in Fig. 1

and ‘Arbeitsgang’ are necessarily also influenced by ‘Betriebsmittelgruppe’.

On the other hand the attributes ‘Losgröße’, ‘Arbeitsgang’, and ‘Arbeitsgangfolge’ are independent. This can be derived from the visualization in Fig. 2, where these three concepts span a three-dimensional cube.

4.2 Supporting the Maintenance of Ontologies

In this section we discuss two ways how Formal Concept Analysis can support the maintenance of ontologies. The first scenario is the creation of an ontology from scratch, the second deals with the addition of new concepts to an existing ontology. In both cases we restrict ourselves to the IS_A relation.

4.2.1 Creating Ontologies from Scratch

The first scenario is based on the assumption that there exists a list of objects of the domain (products, documents, web pages, etc.) and attributes describing them. Then Formal Concept Analysis can be used to compute a concept lattice for these data. The formal concepts are used as concepts of the new-built ontology. It only remains to provide useful names for the concepts. The attributes provide default names for the concepts labeled with them. The names for the other concepts have to be determined by the user in an interactive dialogue.

From the resulting ontology we know that it is always a lattice which means that for all pairs of concepts there always exist a unique least common superconcept and a unique greatest common subconcept. This has two advantages. Firstly, it allows us to compute with concepts, similar to the calculus with least common multiples and greatest common dividers of natural numbers. One consequence is that one can read implications between the attributes (and the ontology concepts) in the concept lattice; another is the fact that the visualization like in Figure 2 is always possible with never more than one label for each object and each attribute [Wi82].

Secondly we can apply the whole methodology of lattice theory for operating with such ontologies. In particular, lattice theory provides us with different operations for dividing, factorizing, and constructing lattices.

Formal Concept Analysis can also deal with many-valued attributes, as for instance found in relational databases. The technique applied is called *conceptual scaling*. It is described in more detail in [GW99]. A first application of conceptual scaling to ontologies is described in [St99].

4.2.2 Extending Given Ontologies to Lattices

Formal Concept Analysis can also be applied to ontologies when no instances of the ontologies (i. e., no objects) are given. In this scenario, Formal Concept Analysis is used for suggesting new concepts to the user. For any given ontology with the only condition that the IS_A relation verifies the axioms of a partial order (which is almost always fulfilled), the resulting ontology is a lattice whenever the user accepts all suggestions. Furthermore it is the smallest lattice in which the original ontology can be embedded, thus it contains no unnecessary concepts and offers the advantages described above.

The technique applied is the *Dedekind-MacNeille-completion* [Ma37] which was first described in terms of Formal Concept Analysis in [Wi82]. It says that, for each partially ordered set $(P, \leq)$, the concept lattice $\underline{\mathfrak{B}}(P, P, \leq)$ is the smallest lattice in which the partially ordered set can be embedded. Applied to ontologies it means that, for a given ontology with a set C of concepts and a partial order IS_A on C, we compute the concept lattice $\underline{\mathfrak{B}}(C, C, \text{IS\_A})$. We suggest all concepts which are in the concept lattice but not in the source ontology as new concepts to the user, and ask him to provide new names for them.

Example. Figure 3 shows part of an ontology in the logistics domain. It is not a lattice, since for instance the two concepts StaudammJangtse and Kernkraftwerk Hokkaido have two minimal common superconcepts (see Figure 4), namely Grosskunde and Auslandskunde. The Dedekind-MacNeille-completion inserts a new concept in between (see Figure 5), which is then named by the user. He may for instance name it "Grosskunde Ausland". As most ontologies have basically tree

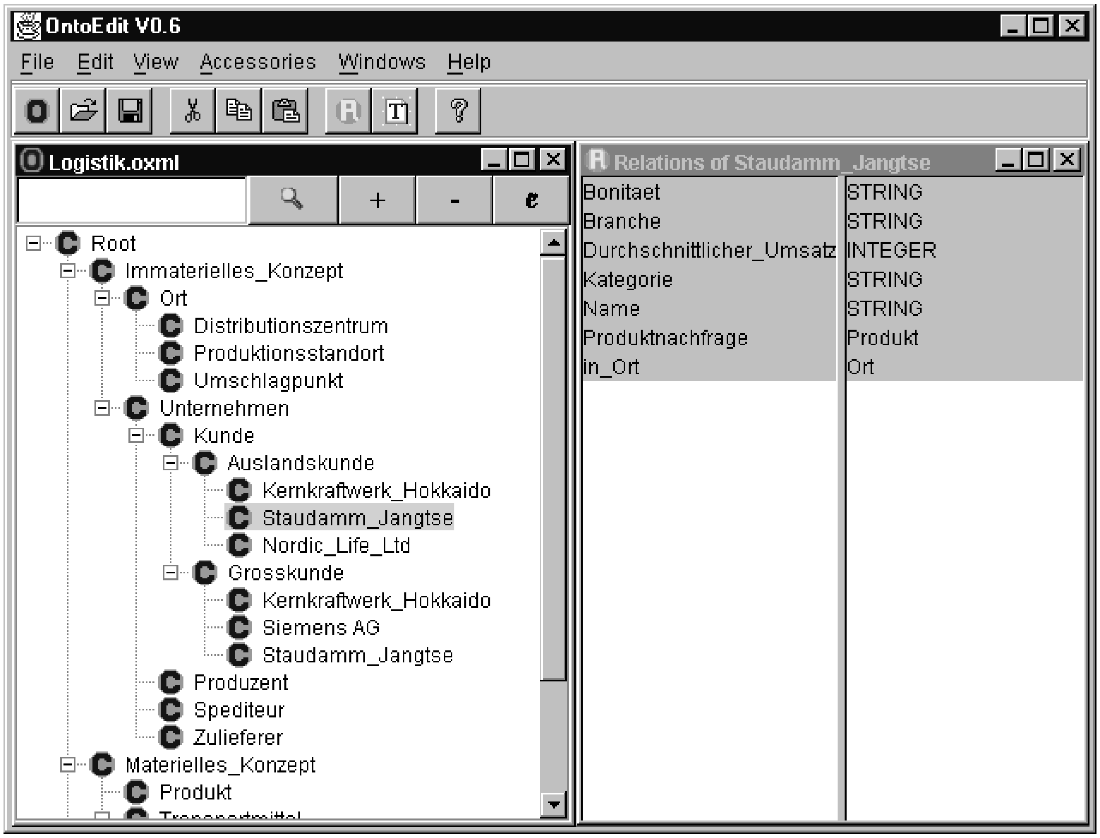

Figure 3: An ontology about logistics

structures, the Dedekind-MacNeille-completion provides only few additional concepts. Hence the interactive dialogue for determining new names will be kept small in general.

5 Outlook

In this paper we have presented a list of research questions about a structural approach to the maintenance of ontologies. Our future work in this project will deal with solutions to the questions raised. One of the topics to keep in mind is the trade-off between benefits from the structural approach on one hand and its rigidness on the other hand. Further research is needed on how to close this gap. The results shall be applied to a collection of ontologies (to be built) in the domain of business processes.

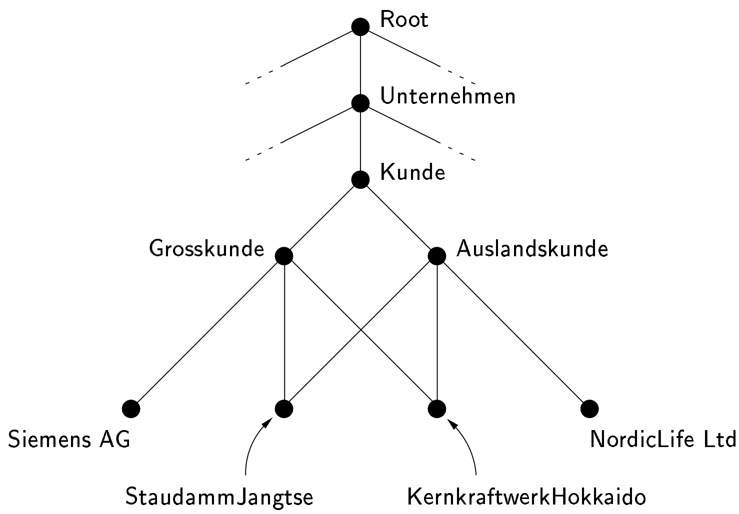

Figure 4: "Customer" part in the logistics ontology

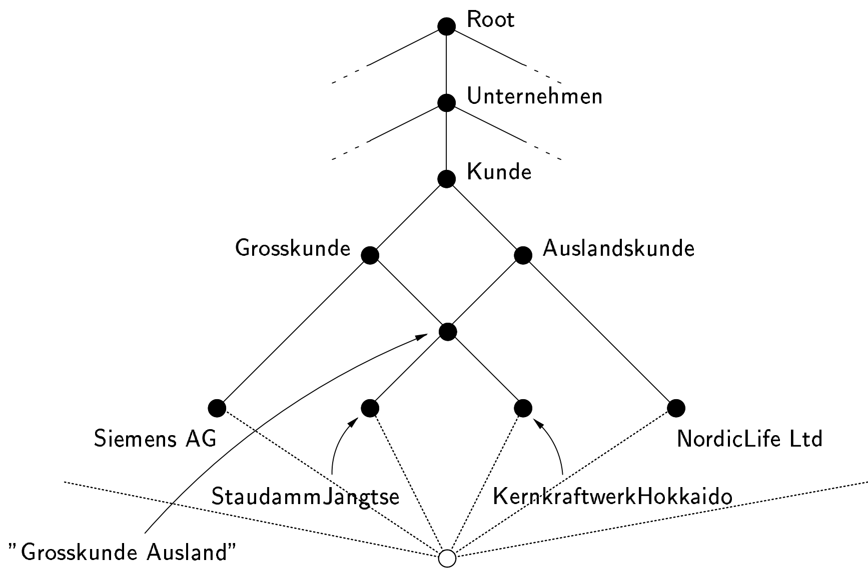

Figure 5: Dedekind-MacNeille-completion of the "customer" part. (The new bottom concept arises out of theoretical reasons and can be ignored in practice.)

Acknowledgement

We wish to thank all members of our research group for lively discussions during the preparation of this work.

References

[AN68] A. Arnauld, P. Nicole: *La logique ou l'art de penser — contenant, outre les règles communes, plusieurs observations nouvelles, propres à former le jugement*. Ch. Saveux, Paris 1668

[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.

[C+98] V. Chaudri, A. Farquhar, R. Fikes, P. Karp, J. Rice: OKBC: A Programmatic Foundation for Knowledge Base Interoperability. *Proc. 15th Natl. Conf. on Artificial Intelligence (AAAI'98)*, Madison, WI, July 1998.

[D+99] S. Decker, M. Erdmann, D. Fensel, R. Studer: Ontobroker: Ontology-based access to distributed and semi-structured information. In: R. Meersman et al. (eds.), Database Semantics: Semantic Issues in Multimedia Systems, Proceedings TC2/WG 2.6 8th Working Conference on Database Semantics (DS-8), Rotorua, New Zealand, Kluwer Academic Publishers, Boston, 1999.

[GW99] B. Ganter, R. Wille: *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin–Heidelberg 1999

[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

[Gu97] N. Guarino: Understanding, building and using ontologies. *Intl. J. of Human and Computer Studies* **46**(2/3), 1997, 293–310

[HSW97] G. van Heijst, A. Th. Schreiber, B. J. Wielinga: Using explicit ontologies in KBS development. *Intl. J. of Human and Computer Studies* **46**(2/3), 1997, 183–292

[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.

[KLW95] M. Kifer, G. Lausen, J. Wu: Logical foundations of object-oriented and frame-based languages, *Journal of the ACM* **42**, 1995

[Ma37] H. M. MacNeille: Partially ordered sets. *Trans. Amer. Math. Soc.* **42**, 1937, 416–460

[MS00a] A. Maedche, S. Staab: Semi-automatic engineering of ontologies from Text. *Proc. 12th Intl. Conf. on Software Engineering and Knowledge Engineering (SEKE'2000)* (to appear)

[MS00b] A. Maedche, S. Staab: Discovering conceptual relations from text. *Proc. 14th European Conference on Artificial Intelligence (ECAI 2000)*, IOS Press, Amsterdam, 2000. (to appear)

[Mc+00] 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.

[Mi+00] P. Mitra, M. Kersten, G. Wiederhold: A graph-oriented model for articulation of ontology interdependencies. *Proc. 7th Intl. Conf. on Extending Database Technology (EDBT 2000)* Springer, Berlin–Heidleberg 2000 (to appear)

[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.

[Sch97] A.–W. Scheer: *Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse.* Springer, Heidelberg-Berlin 1997

[SM00] S. Staab, A. Mädche: Ontology engineering beyond the modeling of concepts and relations. *Proc. ECAI'2000 workshop on application of ontologies and problem-solving methods*, IOS Press, Amsterdam 2000

[St99] G. Stumme: Hierarchies of conceptual scales. *Proc. Workshop on Knowledge Acquisition, Modeling and Management*, Vol. 2. Banff, October 16–22, 1999, 5.5.1–18

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