Using Ontologies and Formal Concept Analysis for Organizing Business Knowledge

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. Ontologies provide explicit models of shared conceptualizations, and are useful for modeling formal, semi-formal, and informal knowledge needed for communication within a company. As there is typically more than one ontology used in a company (for instance one in each department), techniques are needed to merge ontologies in order to assure cross-company communication. In this paper, we summarize one such approach, called FCA–MERGE. During its development, it turned out that a formal definition of 'ontology' is needed. The resulting definition is presented for the first time in detail in this paper.[1]

1 Introduction

Ontologies have been established for knowledge sharing and are widely used as a means for conceptually structuring domains of interest. They have turned out to be a successful approach for structuring informal, semi-formal, and formal knowledge. Ontologies provide an explicit model of a shared conceptualization of some community of interest. Therefore ontologies are a promising formalism for supporting communication with companies and organizations. However it will not be possible in general to agree on one company-wide ontology, as the divergence in the aims of the different players in the company is usually too large. A natural way is to have several ontologies in a company, for instance one for each division, or even one for each department. 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 (e. g., by different departments of a company). In order to support overall communication, methods are needed to bring together these local ontologies, in order to allow for global (cross-department) communication.

In the DFG-project 'OntoWise – Wissensmanagement mit multiplen Ontologien', we have developed FCA–MERGE, an algorithm for merging ontologies, which offers a global structural description of the merging process. The algorithm is based on Formal Concept Analysis, a mathematical theory formalizing the concept of 'concept' ([Wi82,GW99], see also [Wi01]).

[1] This work has been financed by the Deutsche Forschungsgemeinschaft (DFG) under grant STU 170/11–1.

Related work often approaches the task of ontology merging 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. The FCA–MERGE algorithm was presented at the 17th International Joint Conference on Artificial Intelligence (IJCAI '01), see [StuM01]. In this paper, we summarize the results.

Our first attempts to provide a formalization for merging ontologies showed that, at a 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. More general one can say that the commitment to a specific formal definition will influence all future development in the domain.

Therefore, a first step in the project was to develop a precise definition of what we understand under 'ontology'. We emphasize on a mathematical definition which allows to describe the semantics of all ontology operations explicitly. In the definition, we deliberately abstract from implementation details. Those latter are subject to current research within the development of the Karlsruhe Ontology Tool Suite KAON.

1.1 Organization of the Paper

In the next section, we discuss the problems of modeling business knowledge with ontologies. Section 3 gives a summary of our merging approach, and motivates the need for a formal definition of ontologies. In Section 4, we provide the definition of 'ontology' as developed within the project. Section 5 concludes the paper.

2 Business Knowledge and Ontologies

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 are a means 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 (see Figure 1). 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.

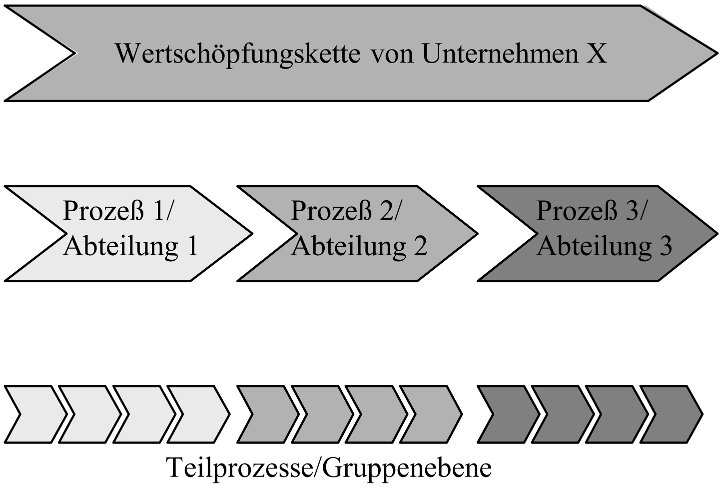

Fig. 1. The hierarchy of departments/sub-processes of a company or organization

A solution to this problem is to provide multiple ontologies which are organized along the organizational structure of the company (departments, working groups, etc.) and/or the structure of the business processes (see Figure 2). 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. In this paper, we focus on the task of ontology merging. Our approach will be discussed in the next section.

3 Ontology Merging Based on Formal Concept Analysis

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

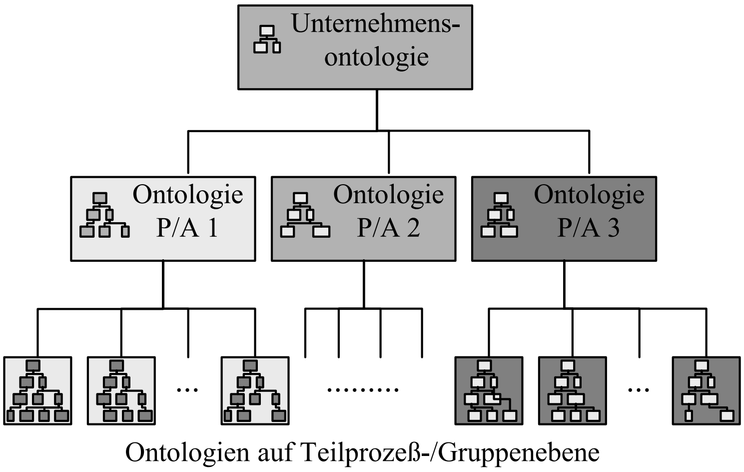

Fig. 2. Ontologies organized along the organizational structure of the company/organization

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.

FCA–MERGE is a method for merging ontologies following a bottom-up approach and offering a global structural description of the merging process. 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 [STBPL01] 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. The approach is described in detail in [StuM01].

FCA–MERGE is based on application-specific instances of the input ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ that are to be merged. The overall process of merging two ontologies is depicted in Figure 3 and consists of three steps, namely *(i)* instance extraction and computing of two formal contexts $\mathbb{K}_1$ and $\mathbb{K}_2$, *(ii)* the FCA-MERGE core algorithm that derives a common context and computes a concept lattice, and *(iii)* the generation of the final merged ontology based on the concept lattice.

The method takes as input the two ontologies and a set D of natural language documents. The documents have to be relevant to both ontologies, so

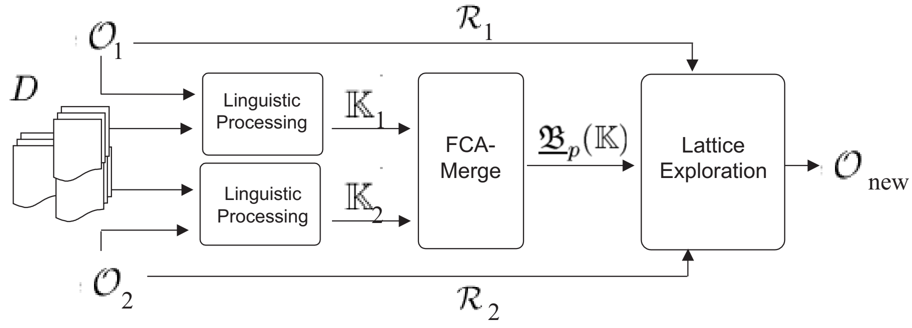

Fig. 3. Ontology Merging Method

that the documents are described by the concepts contained in the ontology. The documents may be taken from the target application which requires the final merged ontology. From the documents in D, we *extract instances*. This automatic knowledge acquisition step returns, for each ontology, a formal context indicating which ontology concepts appear in which documents.

The extraction of the instances from documents is necessary because there are usually no instances which are already classified by both ontologies. However, if this situation is given, one can skip the first step and use the classification of the instances directly as input for the two formal contexts.

The second step of the approach comprises the FCA–Merge core algorithm. The core algorithm merges the two contexts and computes a concept lattice from the merged context using the Titanic algorithm. More precisely, it computes a *pruned concept lattice* which has the same degree of detail as the two source ontologies.

Instance extraction and the FCA–Merge core algorithm are fully automatic. The final step of *deriving the merged ontology* from the concept lattice requires human interaction. Based on the pruned concept lattice and the sets of relation names $\mathcal{R}_1$ and $\mathcal{R}_2$, the ontology engineer creates the concepts and relations of the target ontology. The ontology engineering environment OntoEdit provides graphical means for supporting this process.

Certainly, high quality results of the merging process will always need some 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 a heuristics-based approach as a reliable building block.

As the aim of our approach was to provide a structural description of the merging process, it soon became clear that we would need formal definitions of all structures involved. Hence a first step in the project was to come up with a precise, explicit definition of our understanding of ontologies. This formalization is presented in the next section.

4 The Karlsruhe Perspective on Ontologies

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]. A 'conceptualization' refers to an abstract model of some phenomenon in the world by identifying the relevant concept of that phenomenon. 'Explicit' means that the types of concepts used and the constraints on their use are explicitly defined. This definition is often extended by three additional conditions: "An ontology is an explicit, *formal* specification of a *shared* conceptualization *of a domain of interest*". 'Formal' refers to the fact that the ontology should be machine readable (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 as a group. The reference to 'a domain of interest' indicates that for domain ontologies one is not interested in modeling the whole world, but rather in modeling just the parts which are relevant to the task at hand.

Common to all these definitions is their high level of generalization, which is far from a precise mathematical definition. 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 a modular way we consider first the common core of all ontologies, and then introduce different extensions. Our approach is independent of a specific logical language. It can for instance be used with F-Logic [KLW95], as it is e. g. implemented in Ontobroker [D+99] and OntoEdit [StaM00], but is open to other languages.

Definition 1. *A* core ontology *is a structure*

$$\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R)$$

consisting of

- *two disjoint sets C and R whose elements are called* concept identifiers *and* relation identifiers*, resp.,*
- *a partial order $\leq_C$ on C, called* concept hierarchy *or* taxonomy*,*
- *a function $\sigma\colon R \to C^+$ called* signature*,*
- *a partial order $\leq_R$ on R, called* relation hierarchy*, where $r_1 \leq_R r_2$ implies $|\sigma(r_1)| = |\sigma(r_2)|$ and $\pi_i(\sigma(r_1)) \leq_C \pi_i(\sigma(r_2))$, for each $1 \leq i \leq |\sigma(r_1)|$.*

Often we will call concept identifiers and relation identifiers just *concepts* and *relations*, resp., for sake of simplicity. Almost all relations in practical use are binary. For those relations, we define their *domain* and their *range*.

Definition 2. *For a relation* $r \in R$ *with* $|\sigma(r)| = 2$*, we define its* domain *and its range by* $\text{dom}(r) := \pi_1(\sigma(r))$ *and* $\text{range}(r) := \pi_2(\sigma(r))$.

If $c_1 \leq_C c_2$*, for* $c_1, c_2 \in C$*, then* c_1 *is a* subconcept of c_2*, and* c_2 *is a* superconcept of c_1*. If* $r_1 \leq_R r_2$*, for* $r_1, r_2 \in R$*, then* r_1 *is a* subrelation of r_2*, and* r_2 *is a* superrelation of r_1*.*

If $c_1 <_C c_2$ *and there is no* $c_3 \in C$ *with* $c_1 <_C c_3 <_C c_2$*, then* c_1 *is a* direct subconcept of c_2*, and* c_2 *is a* direct superconcept of c_1*. We note this by* $c_1 \prec c_2$*.* Direct superrelations *and* direct subrelations *are defined analogously.*

Relationships between concepts and/or relations as well as constraints can be expressed within a logical language. We provide a generic definition, which allows the use of different languages.

Definition 3. *Let* $\mathcal{L}$ *be a logical language. A* $\mathcal{L}$-axiom system *for an ontology* $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R)$ *is a pair*

$$A := (AI, \alpha)$$

where

- AI *is a set whose elements are called* axiom identifiers *and*
- $\alpha \colon AI \to \mathcal{L}$ *is a mapping.*

The elements of $A := \alpha(AI)$ *are called* axioms*.*

An ontology with $\mathcal{L}$-axioms *is a pair*

$$(\mathcal{O}, A)$$

where $\mathcal{O}$ *is an ontology and* A *is a* $\mathcal{L}$*-axiom system for* $\mathcal{O}$*.*

In the sequel, *ontology* stands for either a core ontology or an ontology with $\mathcal{L}$-axioms.

According to the international standard ISO 704, we provide names for the concepts (and relations). Instead of 'name', we call them 'sign' to allow for more generality.

Definition 4. *A* lexicon *for an ontology* $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R)$ *is a structure*

$$Lex := (S_C, S_R, Ref_C, Ref_R)$$

consisting of

- *two sets* S_C *and* S_R *whose elements are called* signs for concepts *and* relations*, resp.,*

– *a relation* $Ref_C \subseteq S_C \times C$ *called* lexical reference for concepts, *where* $(c, c) \in Ref_C$ *holds for all* $c \in C \cap S_C$.
– *a relation* $Ref_R \subseteq S_R \times R$ *called* lexical reference for relations, *where* $(r, r) \in Ref_R$ *holds for all* $r \in R \cap S_R$.

Based on Ref_C*, we define, for* $s \in S_C$*,*

$$Ref_C(s) := \{c \in C \mid (s, c) \in Ref_C\}$$

and, for $c \in C$*,*

$$Ref_C^{-1}(c) := \{s \in S \mid (s, c) \in Ref_C\} \ .$$

Ref_R *and* Ref_R^{-1} *are defined analogously.*

An ontology with lexicon *is a pair*

$$(\mathcal{O}, Lex)$$

where $\mathcal{O}$ *is an ontology and* Lex *is a lexicon for* $\mathcal{O}$.

Ontologies formalize the intentional aspects of a domain. The extensional part is provided by a knowledge base, which contains assertions about instances of the concepts and relations.

Definition 5. *A* knowledge base *is a structure*

$$KB := (C_{KB}, R_{KB}, I, \iota_C, \iota_R)$$

consisting of

– *two sets* C_{KB} *and* R_{KB},
– *a set* I *whose elements are called* instance identifiers *(or* instances *or* objects *for short),*
– *a function* $\iota_C: C_{KB} \to \mathfrak{P}(I)$ *called* concept instantiation,
– *a function* $\iota_R: R_{KB} \to \mathfrak{P}(I^+)$ *called* relation instantiation.

As for concepts and relations, we also provide names for instances.

Definition 6. *An* instance lexicon *for a knowledge base* $KB := (C_{KB}, R_{KB}, I, \iota_C, \iota_R)$ *is a pair*

$$IL := (S_I, R_I)$$

consisting of

– *a set* S_I *whose elements are called* signs for instances,
– *a relation* $R_I \subseteq S_I \times I$ *called* lexical reference for instances.

A knowledge base with lexicon *is a pair*

$$(KB, IL)$$

where KB *is a knowledge base and* IL *is an instance lexicon for* KB.

When a knowledge base is given, we can derive the extensions of the concepts and relations of the ontology, based on the concept instantiation and the relation instantiation.

Definition 7. *Let* $KB := (C_{KB}, R_{KB}, I, \nu_C, \nu_R)$ *be a knowledge base. The* extension $[\![c]\!]_{KB} \subseteq I$ of a concept $c \in C$ *is recursively defined by the following rules:*

- $[\![c]\!]_{KB} \leftarrow \iota_C(c)$
- $[\![c]\!]_{KB} \leftarrow [\![c]\!]_{KB} \cup [\![c']\!]_{KB}$, *for* $c' < c$.
- *the axioms in* A *(if* $\mathcal{O}$ *is an ontology with* $\mathcal{L}$*-axioms).*

The extension $[\![r]\!]_{KB} \subseteq I^+$ of a relation $r \in R$ *is recursively defined by the following rules:*

- $[\![r]\!]_{KB} \leftarrow \iota_R(r)$
- $[\![r]\!]_{KB} \leftarrow [\![r]\!]_{KB} \cup [\![r']\!]_{KB}$, *for* $r' < r$.
- *the axioms in* A *(if* $\mathcal{O}$ *is an ontology with* $\mathcal{L}$*-axioms).*

If the reference to the knowledge base is clear from the context, we also write $[\![c]\!]$ and $[\![r]\!]$ instead of $[\![c]\!]_{KB}$ and $[\![r]\!]_{KB}$.

The following definition tells us if a knowledge base is consistent with an ontology.

Definition 8. *A knowledge base* $KB := (I, \iota_C, \iota_R)$ *is* consistent with an ontology $\mathcal{O}$, *if all of the following hold:*

- $\mathcal{O}$ *is consistent (if* $\mathcal{O}$ *is an ontology with* $\mathcal{L}$*-axioms),*
- $C_{KB} \subseteq C$,
- $R_{KB} \subseteq R$,
- $[\![r]\!] \subseteq \prod_{c\in\sigma(r)} [\![c]\!]$, *for all* $r \in R$
- KB *is a model for* $A \cup \{c_1 \leq c_2 \rightarrow [\![c_1]\!] \subseteq [\![c_2]\!] \mid c_1, c_2 \in C\} \cup \{r_1 \leq r_2 \rightarrow [\![r_1]\!] \subseteq [\![r_2]\!] \mid r_1, r_2 \in R\}$.

5 Conclusion

In this paper, we have analyzed the problems of modeling business knowledge with ontologies. We have summarized FCA–Merge, an algorithm for ontology merging, which provides a structural description of the merging process. During its development the need of a formal definition of what we understand under 'ontology' arose. The resulting formal definition is presented in this paper for the first time in detail.

Acknowledgement

Many thanks go to all members of our group, who contributed to the formalization of our understanding of 'ontology' in a series of lively discussions.

References

[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

[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, Heidelberg 1999. (Translation of: *Formale Begriffsanalyse: Mathematische Grundlagen.* Springer, Heidelberg 1996.)

[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

[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

[NM00] N. F. 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

[StaM00] 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

[StuM01] 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

[STBPL01] G. Stumme, R. Taouil, Y. Bastide, N. Pasqier, L. Lakhal: Computing Iceberg Concept Lattices with Titanic. *J. on Knowledge and Data Engineering* (to appear)

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

[Wi01] R. Wille: Begriffliche Wissensverarbeitung: Theorie und Praxis. *Informatik Spektrum* **23**(6), 2000, 357–369