A Meta-ontological Architecture for Foundational Ontologies

Heinrich Herre and Frank Loebe

Research Group Ontologies in Medicine (Onto-Med),
Institute of Medical Informatics, Statistics, and Epidemiology (IMISE) and
Institute of Informatics (IfI), Leipzig University, Germany
{herre,loebe}@informatik.uni-leipzig.de
http://www.onto-med.de

Abstract. In this paper we present and discuss a meta-ontological architecture for ontologies which centers on *abstract core ontologies* (ACOs). An ACO is the most abstract part of a foundational ontology. It is useful for an ontologically founded description of ontologies themselves, therefore ACOs are lifted to the meta-level. We propose a three-layered meta-ontological architecture which distinguishes an object level comprising foundational, generic or domain-specific ontologies, a meta-level with abstract core ontologies, and a meta-meta-level employing *abstract top ontologies* for the formalization of the underlying levels. Moreover, two axiomatic fragments for ACOs are provided, one of which is applied to *formal concept lattices* [1]. This demonstrates the use of ACOs for the ontological foundation of representation formalisms and illustrates advantages in comparison to the usual direct formal reduction to set theory. Finally, related work with respect to the architecture is briefly discussed.

1 Introduction

There is a rapidly growing body of work on ontologies in information systems over the last 10 to 15 years, which has been boosted by the vision of the Semantic Web. Likewise, research in formal tools and techniques related to ontology development (or *ontological engineering*) is very active. By Grubers definition of ontologies as sharable conceptual specifications [2], their development is an issue closely related to the field of conceptual modeling.

From the very beginning, ontologies were distinguished according to their intended range of applicability. In particular, *foundational ontologies*[1] were considered to provide the most general kinds of entities as a basis for more specific ontologies. However, work on foundational ontologies has resulted in rather large and complex systems. This is problematic if foundational ontologies are to be applied to identify and express ontological commitments of representation formalisms.

[1] Also referred to as *top-level ontologies* in the literature; examples are: DOLCE [3], GFO [4], SUMO [5], and Seibt's [6], Sowa's [7] and West's [8] ontologies

R. Meersman and Z. Tari (Eds.): CoopIS/DOA/ODBASE 2005, LNCS 3761, pp. 1398–1415, 2005.

Furthermore, like conceptual models ontologies face the problem of meta-modeling, i.e., the question of which basic vocabulary should be used to define and explain ontologies, especially their entities as well as constraints and logical interdependences among them. Of course, there are already efforts like [9], many of which are inspired by meta-modeling experience in conceptual modeling. Nevertheless, we intend to contribute to a clear and comprehensible meta-architecture for ontologies. This is largely due to the fact that most ontology languages are directly reduced to an underlying set-theoretical model in order to provide a formal semantics, which especially holds for logic-based formalisms like the Web Ontology Language (OWL, [10]).

The aim of this paper is to provide a clear analysis of meta-language aspects of ontologies and ontology languages in order to develop a comprehensible, yet comprehensive meta-ontological architecture. In particular, a clearly arranged core of entity types is to be extracted from current foundational ontologies, which will be called an *abstract core ontology* (ACO). On the one hand, ACOs are to be used as a meta-level for ontologies. On the other hand, they can be employed for the ontological foundation of representation formalisms in an easier way compared to rich foundational ontologies.

According to these aims, the organization of the paper is as follows. In section 2 we analyze the use of meta-languages in general as well as the role of set theory for modeling and modeling formalisms. On this basis our meta-ontological architecture is presented. Section 3 introduces the central notion of ACOs in detail. First, a number of entities and relations for ACOs are identified and informally described. Following an approach which admits variants of ACOs, two particular ACOs are presented as formal fragments. The first of these is applied to Formal Concept Analysis (FCA) [1] in section 4. The final section 5 concludes the paper with a summary, a brief comparison with related work, and future directions.

2 Meta-ontological Analysis and Architecture

For every formalism there is the need to explain the relationship between its syntax and semantics. In particular, this task comprises the determination of the ontological commitments of the formalism itself or in conjunction with its application to a certain task. Thus, we start with considerations of meta-languages.

2.1 Meta-languages

Let W be a world of objects. A formal language $\mathcal{L}$ whose expressions refer to the objects in W is called an *object-level language* for W. In order to specify and communicate the meaning of these expressions, a *meta-language* $\mathcal{M}$ for the pair $(\mathcal{L}, W)$ is required. That means, $\mathcal{M}$ is a language whose expressions refer to the items included in $\mathcal{L}$ or in both, $\mathcal{L}$ and W, but which also refer to relations between $\mathcal{L}$ and W. A formal language $\mathcal{L}$ has a semantics if there is a class Sem of objects and a relation $\mathsf{den}(x, y)$ relating expressions of $\mathcal{L}$ to the objects of Sem.

The denotation relation $\mathsf{den}(x, y)$ stipulates a connection between a symbol x and a semantic object y.

The first point to note is that the notions of symbol and denotation are at the heart of the transition from informal to formal languages, but equally relevant for the explanation of natural language semantics. These notions have been puzzling philosophers of logic and language ever since (cf. [11]–[14]), but herein we restrict to a simplified view.

According to this, one may assume a basic relation which associates the symbols of a language to the objects of the real world. For example, we may say that the phrase "the moon" denotes a certain real object in the sky. We take denotation as an interface between the informal and the formal treatment of an ontology. Therefore, the notions around denotation are used in informal explanations of meta-language aspects, but in the formal treatment of meta-levels these entities are not taken into account. Further, it is assumed that only natural language can be used as a meta-language for any kind of language, including itself. However, an infinite regress arises if every expression is to be defined within language. For instance, if one claims that the word "moon" denotes a certain real thing in the sky, then the phrase "a certain real thing in the sky" is another expression which requires a certain meaning. This regress can only be avoided if we assume that there is an original anchoring relation relating symbols to objects, which has to be assumed as a basic intuition without further specification in language, neither formal nor informal.

2.2 The Role of Set Theory

Set theory is a convenient mathematical tool to describe and model things and structures. One may speak about sets (collections) of things, about graphs, algebras, operations etc. Sets are abstract atemporal entities; for example, considering a set $\{a\}$ of an apple a does not change anything on that apple. Mathematical objects may be founded on set theory, and hence, in describing parts of the world we may construct a set-theoretical structure which is associated to these parts of the world and which *models* them. For describing the language of set theory we need an appropriate meta-language; to simplify the matter we assume that in any case natural language is available as a (non-formal) meta-language. As is well-known, the meta-account of set theory includes the notions of *sets*, *urelements*, and the *membership* relation.

Moreover, set theory is intimately tied to logical languages due to the fact that the commonly accepted approach of Tarski-style model-theoretic semantics (cf. [15]) is based on set-theoretical constructions. The relationship between such languages and their meta-theoretical treatment is well established. At present many ontology languages are logical languages with model-theoretic semantics (cf. [10, 16, 17, 18]). Therefore, a sufficiently rich fragment of set theory should provide an appropriate basis for a formal account of a meta-language, i.e., set theory then serves as a meta-meta-language for object-level ontologies.

On the other hand, we do not intend to restrict ourselves to set theory alone, of which distinct variants exist. Instead, a more generic route is taken in the next section, generalizing the role of set theory as a meta-meta-language for ontologies to the notion of an abstract top ontology. This allows for other formalisms to be used as the meta-meta-account. For example, a minor deviation from set theory would be to consider variants of it, like hypersets [19]. But more radical choices are conceivable, for instance in favor of *mathematical category theory* [20]. In every case, any of these have to be chosen with great care as they lay the formal foundation for analyses of ontologies.

2.3 Meta-ontological Architecture

In our work on the General Formal Ontology (GFO, cf. [4]), which is a foundational ontology, the need for a meta-ontological level arose, which is not already a set-theoretical reduction. In striving towards such an approach we emphasize generality and comprehensibility as requirements. Note that herein we focus on a basic vocabulary for entities within ontologies, whereas ontologies as a whole or complex components of ontologies are not discussed.

In figure 1 a three-layered meta-ontological architecture is proposed. As introduced in section 2.1, natural languages form a universally applicable, but informal approach to every level. The three layers on the right-hand side of figure 1 provide a well-founded formal approach.

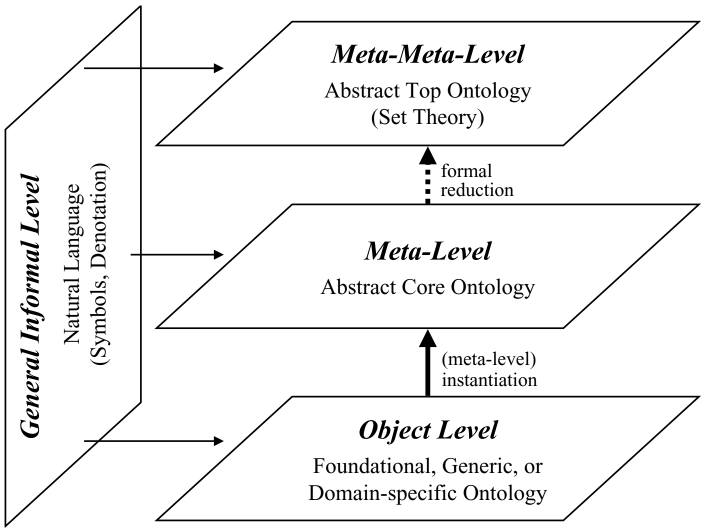

Fig. 1. Meta-ontological Architecture

The lowest level refers to the objects of the meta-architecture, viz. to ontologies with varying degrees of their range of applicability. The meta-level provides the notions to work with object-level ontologies. In section 3.1 the most general basic entities are specified which can be included in the meta-level. Based on these notions several abstract core ontologies may be constituted, which are determined by a selection of basic notions and by axioms stated about them. Hence we admit different variants of ACOs, two of which are presented in sections 3.2 and 3.3. In this respect our work is inspired by conceptual modeling, cognitive linguistics, philosophy, and by our work on the foundational ontology GFO, expounded in [4].

In order to be able to treat ACOs in a formal manner, there is a need for a formal meta-account of them. This leads to the introduction of the notion of an *abstract top ontology* (ATO), which generalizes the role of set theory as described in section 2.2.

Often set theory itself is viewed to take the role of an ACO instead of an ATO. However, our experience shows that in many cases representation languages are ontologically richer than set theory (cf. [21]). Hence the introduction of an intermediate layer appears reasonable. Looking at ACOs from the opposite side, namely from the object level, formerly the notion of foundational ontologies was considered to fill this role already. However, the growing body of work on foundational ontologies shows that these are too rich already to integrate all their features into formalisms directly. Thus, we introduce the distinction between ACOs and foundational ontologies.

Some remarks on the *relationships* between object, meta- and meta-meta-level need to be made. Frequently, the relations between these levels are considered to be of the same character, often a vague notion of instantiation (cf. section 5.2). This is not the case for the levels above. From our point of view, the ontological notion of instantiation is preserved only between the object and the meta-level, since they provide ontological content. However, the transition from the ACO level to the ATO level is different. For example, in the case of set theory as an ATO one may speak of a set-theoretical reduction of the entities on the meta-level. Again, the mere intention behind the ATO level is to provide a rigorous mathematical framework in order to study formal properties of ontologies on the lower levels.

These issues may become clearer by example. In section 3.1 several entities and relations will be introduced, among them the notion of *category* and (object-level) *instantiation*. These belong to the meta-level, i.e., they are ontological entities whose interconnections to other entity types need to be clarified. Later then, when it comes to formalizing these interconnections, these notions are expressed in the vocabulary of the meta-meta-level. More precisely, category will be modeled as a set Cat whose elements are set-theoretical urelements, whereas instantiation is modeled as a set denoted by $::$ with pairs of urelements as its elements. In contrast to this reduction on the meta-meta-level, the relation between meta- and object level is different. Categories and instantiation are not merely "implementation devices" for ontologies. In fact, it is debatable whether some or even all of the meta-entities should be reflected on the object level. However, in contrast to a fully self-reflective ontology and for the sake of clarity and simplicity, the separation into meta- and object level is preferred herein (cf. also [22]).

3 Abstract Core Ontologies (ACOs)

This section elaborates the meta-level of the architecture which is formed by abstract core ontologies. Let us first introduce and discuss the pragmatic and functional aspects of ACOs. An abstract core ontology captures such entities which are built – implicitly or explicitly – into the semantics of a wide number of general representation formalisms and which commonly occur in foundational ontologies. Examples for foundational ontologies are given in section 1, hence it remains to clarify the term "general representation formalism". It refers to such formalisms which are not adapted to particular domains of discourse, but which can be applied to a broad scope of modeling and representation problems. Examples range from first-order logic over Semantic Web languages like the Web Ontology Language (OWL, [10]), frame-based approaches as in [23], to modeling languages like UML [24].

Apart from pragmatic aspects, ACOs need to be determined firstly by their main entity types and relations among them, for which a certain vocabulary is proposed below. Secondly, logical interdependences of those entities and relations need to be specified. After an informal description of these connections formal accounts are elaborated in sections 3.2 and 3.3. The latter exemplify the formalization of several types of interdependences by axioms of first-order logic. Note that this approach relies on the meta-meta level, where in our case set theory is employed as the abstract top ontology.

3.1 Entities for ACOs

In identifying entities for ACOs we pursue a pragmatic approach which is further based on philosophical considerations, in particular on the work of J. Gracia [25, 26]. We will not insist on a single ACO, but we specify a set of entities (cf. table 1), of which certain subsets may suffice for a particular ACO.

Before we go into details of this vocabulary, let us once more mention the strict separation between the levels introduced in section 2.3. We stipulate that none of the meta-level relations are available as objects on the object level. This requirement implies that infinite regresses like Bradley's infinite regress of instantiation (cf. [27]) can be avoided. The same requirement is assumed for most of the meta-level entities, although reflections of them on the object level may be less complicated. For example, one could include the meta-level entity "Individual" also as an entity of the object level.

Our starting point is that the entities of the world may be divided into *categories* and *individuals*, i.e., everything in an ontology is either a category or an individual. Categories are entities which may be predicated of or *instantiated by* other entities; in the opposite, individuals essentially cannot be instantiated or predicated of other entities [26]. Note, however, that there are also categories which cannot have instances, e.g. due to their logical structure, like a "round square". Nevertheless, such categories are distinct from individuals. The most natural individuals are related to time and space, whereas categories are atemporal, abstract entities.

Table 1. Basic Entity Types and Relations

Meta-Level Entity Types (Sets of urelements)			
Name	*Symbol*	*Name*	*Symbol*
Category	Cat	Individual	Ind
Object Category	OCat	Object	Obj
Property	P	Attribute	Att
Role Category	RCat	Role	Rol
Relation	R	Relator	Rel

Meta-Level Relations (Sets of pairs of urelements)		
Name	*Symbol*	*Argument Restrictions*
identity	$x = y$	–
instantiation	$x :: y$	$\mathsf{Cat}(y)$
inherence	$\mathsf{inh}(x, y)$	$\mathsf{Att}(x)$ or $\mathsf{Rol}(x)$
role-of	$\mathsf{role}(x, y)$	$\mathsf{Rol}(x)$, $\mathsf{Rel}(y)$
categorial part-of	$\mathsf{catp}(x, y)$	$\mathsf{Cat}(x)$, $\mathsf{Cat}(y)$

Moreover, among individuals we distinguish *objects*, *attributes*, *roles* and *relators*. Objects are "complex" entities which have attributes and which play certain roles in respect to other entities. Objects are to be understood as similarly general as the notion of "object" in object-oriented analysis. In particular, objects comprise animate and inanimate things like humans, trees or cars as well as processes like this morning's sunrise[2]. Examples of attributes are particular weights, forms, colors, etc. A sentence like "This rose is red." refers to a particular object which is a rose, and to a particular attribute which is a red. Another basic relation is needed to connect objects and attributes. The phrases "having attributes" and "playing a role" above are captured by the basic relation of *inherence*, which is such that an attribute or a role inheres in some object. This relation expresses the dependence of attributes and roles on entities in which they can inhere.[3]

The difference between attributes and roles consists in the fact that roles are interdependent with other roles [28]. Examples of roles are available through terms like parent, child or neighbor. Here, parent and child would be considered as a pair of interdependent roles. Apparently, these examples easily remind one of relations like "is-child-of". Indeed, a composition of interdependent roles is a *relator*, i.e., an entity which connects several other entities. The formation of relators from roles further involves the basic relation of *role-of*.

Thus far we have explained the right-hand side of the upper part of table 1, i.e., the subdivision of individuals and the interrelations among them. However, in explaining roles by reference to terms like parent and child, there was already

[2] More fine-grained distinctions among objects do not belong to the ACO level, but appear as object-level instances of ACO notions, e.g. within some foundational ontology.

[3] This commonality allows one to generalize attributes and roles to *qualities*. Note that this notion of quality differs from the term as used in [4].

a transition to the realm of categories. Actually, parent and child refer to *role categories*. As is obvious from the table, for all specializations of individuals introduced above there are categorial counterparts.

Two basic relations have not been discussed thus far: *identity* and *categorial part-of*. We will not dwell on the intricacies of identity, only to mention that we refer to it as a notion in the interface between natural and formal languages. We expect a full account of identity to involve denotation, possibly along the lines of the Peircean notion of co-reference (cf. [7]). Categorial part-of is more in the focus of the present work. Its arguments are categories; it directly reflects dependencies among categories and uncovers how one category may be constructed out of others. For example, the property of being round may be a categorial part of the category of round glasses. Another example is the category apple, of which one may consider categorial parts like a certain form or color and the like. There are two intricate issues regarding categorial part-of. Firstly, a categorial part must not be misinterpreted to apply to a category directly. For example, a category with the property red as categorial part is not red itself. This problem is easily avoided in the formalization. Secondly, it should be noted that inherence and categorial part-of are irreducible with respect to each other. Section 4 will provide further applications of categorial part-of.

3.2 Categories, Properties, and Objects

In this section we consider an abstract core ontology, denoted by CPO, which is based on categories and properties only, together with their individual counterparts and the corresponding meta-level relations. Object-level relations and role categories as well as relators and roles are not included. Formally this corresponds to the following signature:

$$\Sigma^{CPO} = (\mathsf{Cat}, \mathsf{OCat}, \mathsf{P}, \mathsf{Ind}, \mathsf{Obj}, \mathsf{Att}, =, ::, \mathsf{inh}, \mathsf{catp}) \tag{1}$$

Several languages can be used to formalize such systems. In the following, a type-free first-order language is assumed, although others may be appropriate as well (e.g. typed languages). We present axiomatic fragments pertaining to the signature introduced above.

First of all, implicit assumptions need to be explicated, which are often taken for granted by human readers on the basis of natural language descriptions, like the one given in section 3.1. First, there are disjointness and coverage conditions for entities. Concerning (4), remember that CPO omits relators and roles.

$$\neg\exists x \ \ (\mathsf{Cat}(x) \land \mathsf{Ind}(x)) \tag{2}$$

$$\forall x \ \ (\mathsf{Cat}(x) \lor \mathsf{Ind}(x)) \tag{3}$$

$$\forall x \ \ (\mathsf{Ind}(x) \to \mathsf{Obj}(x) \lor \mathsf{Att}(x)) \tag{4}$$

Meta-level subsumption relations need to be expressed, which are indicated in table 1 by indentation.

$$\forall x \ \ (\mathsf{Obj}(x) \to \mathsf{Ind}(x)) \tag{5}$$

$$\forall x \ \ (\mathsf{P}(x) \to \mathsf{Cat}(x)) \tag{6}$$

Regarding the meta-level relations, similar axioms have to be stated (7). In addition, an account of the argument restrictions from table 1 is needed (8). In an opposite manner, claims of existence can be expressed (9).

$$\neg(\exists xy \ \ (x :: y \wedge \mathsf{inh}(x, y))) \tag{7}$$
$$\forall xy \ \ (\mathsf{inh}(x, y) \rightarrow \mathsf{Att}(x) \wedge \mathsf{Obj}(y)) \tag{8}$$
$$\forall x \ \ (\mathsf{Obj}(x) \rightarrow \exists y(\mathsf{Att}(y) \wedge \mathsf{inh}(y, x))) \tag{9}$$

Moreover, definitions may reveal "reducible" entities or new, definable notions which are also interesting from an ontological point of view. In the case of CPO, categories can be used to define their individual counterparts (10-11), but not vice versa due to the possibility of categories without instances.

$$\forall x(\mathsf{Ind}(x) \leftrightarrow \neg\mathsf{Cat}(x)) \tag{10}$$
$$\forall x(\mathsf{Obj}(x) \leftrightarrow \exists y(\mathsf{OCat}(y) \wedge x :: y)) \tag{11}$$
$$\forall x(\mathsf{Att}(x) \leftrightarrow \exists y(\mathsf{P}(y) \wedge x :: y)) \tag{12}$$
$$\forall x(\mathsf{PrimCat}(x) \leftrightarrow \mathsf{Cat}(x) \wedge \exists y(y :: x) \wedge \forall z(z :: x \rightarrow \mathsf{Ind}(z))) \tag{13}$$
$$\forall x(\mathsf{AttCat}(x) \leftrightarrow \mathsf{Cat}(x) \wedge \exists y(y :: x) \wedge \forall z(\mathsf{catp}(z, x) \rightarrow \mathsf{P}(z))) \tag{14}$$

Note that (10) already follows from (3) and (2). The newly defined notions of a *primitive category* (13) and an *attributive category* (14) are relevant for section 4. They reveal branching points for the ACO. For instance, the question arises whether one should assume that all categories are primitive. On the other hand, categories of higher types may be admitted. Similarly, attributive categories allow for new constraints: the equality of attributive categories may be defined (15), and, as we decided for CPO, attributive categories may be the only admissable categories (16). From the latter follows that catp is constrained to properties in its first argument.

$$\forall xy \ \ (\mathsf{AttCat}(x) \wedge \mathsf{AttCat}(y) \rightarrow (x = y \leftrightarrow \forall z(\mathsf{catp}(z, x) \leftrightarrow \mathsf{catp}(z, y)))) \tag{15}$$
$$\forall x \ \ (\mathsf{Cat}(x) \rightarrow \mathsf{AttCat}(x)) \tag{16}$$

3.3 Categories, Relations, and Objects

The second abstract core ontology herein is denoted by CRO, and it comprises all notions which were introduced in table 1, in particular roles and object-level relations. This yields the following signature:

$$\Sigma^{CRO} = (\mathsf{Cat}, \mathsf{OCat}, \mathsf{P}, \mathsf{RCat}, \mathsf{R}, \mathsf{Ind}, \mathsf{Obj}, \mathsf{Att}, \mathsf{Rol}, \mathsf{Rel}, =, ::, \mathsf{inh}, \mathsf{role}, \mathsf{catp}) \tag{17}$$

To make our approach to relations clear, we insert an example that illustrates the formalization of an object-level relation. On the object level, let John be a parent of Mary. According to the framework described, John and Mary are objects, and there is a relator which instantiates the parent-of relationship and which connects John and Mary. This connection is established via two roles, one of which being a role of John, the other being a role of Mary. Let j, m, po, p, c be constants for John, Mary, the parent-of relationship, and the parent and the child role categories, respectively. Then, the following formalizes the example:

$$\mathsf{Obj}(j) \wedge \mathsf{Obj}(m) \wedge \mathsf{R}(po) \wedge \mathsf{RCat}(p) \wedge \mathsf{RCat}(c) \tag{18}$$

$$\exists q_1 q_2 r (r :: po \wedge q_1 :: p \wedge q_2 :: c \wedge$$
$$\mathsf{inh}(q_1, j) \wedge \mathsf{inh}(q_2, m) \wedge \mathsf{role}(q_1, r) \wedge \mathsf{role}(q_2, r)) \tag{19}$$

Of course, similar axioms as those presented for CPO belong to the axiomatization of CRO, where not all of those specified for CPO can be reused directly. For instance, (4) and (8) need to be modified to include roles. We will not specify these modifications as they are easily found. Rather, we introduce some further axioms pointing to advantages and problems with this role-based account of relations. A discussion of these issues beyond that is available in [28], including an analysis of the advantages of this approach in comparison to viewing relations as having tuples of entities as instances.

$$\forall x \ \ (\mathsf{Rel}(x) \rightarrow \exists yz(y \neq z \wedge \mathsf{Rol}(y) \wedge \mathsf{Rol}(z) \wedge \mathsf{role}(y, x) \wedge \mathsf{role}(z, x))) \tag{20}$$
$$\forall xyz \ \ (\mathsf{inh}(x, y) \wedge \mathsf{inh}(x, z) \rightarrow y = z) \tag{21}$$

Axiom (20) enforces that relators are really mediating entities, i.e., more than one role is involved, which is even the case if an object is related to itself. (21) is known for attributes as the *non-migration principle*. However, note that in CRO it also refers to roles which can inhere in a single entity only. This is not to be confused with an entity in which several roles of the same role category inhere. For example, John can be a parent of Mary as well as of Fred. Another question concerns the arguments of relations, i.e., whether one should restrict the second argument of inherence. For the sake of simplicity one would often restrict it to objects (22).

$$\forall xy(\mathsf{inh}(x, y) \rightarrow \mathsf{Obj}(y)) \tag{22}$$

However, in a more expressive system, relators between other entity types may be required. In that case, it would be a common approach to classify relators according to the types of entities their roles inhere in. An equivalent classification could be done for the corresponding relations. Before we can give an example, we need to enforce that every relation is bound to certain role categories (23-24), (cf. also [28], sect. 3.3.3). Then, for example the parent-of relation can be claimed to satisfy (26).

$$\forall xy \ \ (\mathsf{R}(y) \wedge \mathsf{catp}(x, y) \rightarrow \mathsf{RCat}(x)) \tag{23}$$
$$\forall x \ \ (\mathsf{R}(x) \rightarrow \exists y(\mathsf{catp}(y, x))) \tag{24}$$
$$\forall x \ \ (\mathsf{ObjRCat}(x) \leftrightarrow \forall yz(y :: x \wedge \mathsf{inh}(y, z) \rightarrow \mathsf{Obj}(z))) \tag{25}$$
$$\forall xy \ \ (\mathsf{ObjR}(x) \leftrightarrow \forall y(\mathsf{catp}(y, x) \rightarrow \mathsf{ObjRCat}(y))) \tag{26}$$

The introduction of axiomatic fragments for two ACOs, CPO and CRO, shows that there are many branching points even for these fairly small ontologies. In connection with a formal account of ACOs, these subtleties need to be explicated and related to each other. This is one of the reasons why we expect that it is more appropriate to start with a restricted number of primitives on the ACO level of our meta-ontological architecture.

Apart from the use of ACOs as meta-ontology for object-level ontologies, their application to general representation formalisms has been mentioned. In the next section we show by example how CPO is employed for the ontological foundation of one such formalism, namely *formal concept lattices*. A stronger evaluation of the work presented here in quantifiable terms remains as future work.

4 Formal Concept Lattices

Formal Concept Analysis (FCA, [1]) is a mathematical formalism with a central notion of *concept lattices* which can be used for conceptual data analysis and knowledge processing.[4] It has gained wide spread from its earliest beginnings in the late 1970s. We show that the theory of concept lattices can be interpreted in our framework of abstract core ontologies, thus giving them an ontological foundation. Moreover, concept lattices can be applied in diverse ways which can be clarified by explicit reference to ACOs. Accordingly, we believe that each use of formal concept analysis should be coupled with an appropriate ontological analysis in order to gain clarity and to use the results of these formal techniques adequately.

The following are the basic definitions of FCA [1] to which we refer herein.

Definition (*Formal Context and Concept*)
A formal context $\mathcal{K} = (G, M, I)$ consists of two sets G and M and a relation I between G and M. The elements of G are called the *objects*, and elements of M are called the *attributes* of the context. $I(g, m)$ means that the object g has the attribute m.
For a set $A \subseteq G$ of objects and a set $B \subseteq M$ of attributes, the set of attributes common to A is $Attr(A)$, the set of objects with all attributes in B is $Objt(B)$:

$$Attr(A) = \{m \mid m \in M \text{ and } I(g, m) \text{ for all } g \in A\} \tag{27}$$

$$Objt(B) = \{g \mid g \in G \text{ and } I(g, m) \text{ for all } m \in B\} \tag{28}$$

A *formal concept* of $\mathcal{K}$ is a pair (C, D) with $C \subseteq G$, $D \subseteq M$, and $C = Objt(D)$, $D = Attr(C)$. C is called the *extent* and D the *intent* of the concept (C, D).

This definition is an example for the specification of a formalism by direct reference to the meta-meta-level of our approach, where set theory serves as abstract top ontology in this case. However, the use of terms like attribute, object,

[4] See also `http://www.math.tu-dresden.de/$\sim$ganter/fba.html`

intent and extent exemplifies hidden ontological assumptions. As examples in [1] indicate, there are several possibilities of interpreting the formalism on the ACO level. That means, it can be used with different ACO level readings which we will now analyze. In the sequel it is assumed that CPO is sufficiently expressive for every of these interpretations. The adequacy of this assumption is reconsidered at the end of this section.

In the first interpretation, G is a set of objects (in the sense of section 3.1, i.e., understood as individuals with attributes), M is a set of properties (categories whose instances are attributes). Then the relation I is interpreted by a relation I_1 defined in terms of inherence and instantiation as follows:

$$I_1(g, m) \leftrightarrow \exists x(x :: m \wedge \mathsf{inh}(x, g)) \tag{29}$$

However, in most examples in the applications of concept lattices the objects are not individuals, but themselves categories. For instance, one example states "A frog needs water to live" for an object "frog" and an attribute "need water to live". Then frog is an object category whose instances are individual frogs, where every individual frog needs water to live. "Needs water to live" has to be understood as a property in the terms of CPO. Hence, in the second interpretation of I, denoted by I_2, the expression "A frog needs water to live" reads as "For every instance of the 'category frog' ($g \in G$) there is an instance of the property 'to need water to live' ($m \in M$) which inheres in that individual frog." This ontologically correct reading of that sentence can be formalized as:

$$I_2(g, m) \leftrightarrow \forall y(y :: g \rightarrow \exists x(x :: m \wedge \mathsf{inh}(x, y))) \tag{30}$$

A third interpretation of I is given by I_3, which in the example corresponds to "the category frog has as a categorial part the property 'to need water to live' ":

$$I_3(g, m) \leftrightarrow \mathsf{catp}(m, g) \tag{31}$$

Apparently, it is not the category frog which needs water to live but the individual frogs. Thus, the right-hand side of (31) may entail the right-hand side of (30). In order to achieve this, a suitable ACO should include to following axiom:

$$\forall xy(\mathsf{catp}(x, y) \rightarrow \forall u(u :: y \rightarrow \exists w(w :: x \wedge \mathsf{inh}(w, u)))) \tag{32}$$

Moreover, if the third reading (I_3) is assumed, one may claim that for concept lattices all expressable categories are attributive categories, i.e., axiom (16) of CPO applies. Altogether, CPO provides at least three exact and ontologically adequate interpretations of the relation I defined for concept lattices.

The analysis continues with the question for the nature of the notion of a formal concept (C, D), which depends on the relation between formal objects and attributes. Assuming the third reading (I_3) from above, the extent C of (C, D) is a set of attributive categories, whereas the intent D is a set of properties. More precisely and following the definition of a formal concept, every property in D is a categorial part of each category in C. Ontologically, it seems that (C, D) is best understood as a category $\hat{c}$ with all properties in D as its categorial parts and *subsuming* all categories in C:

$$x \in C \rightarrow \mathsf{AttCat}(x) \wedge \forall y(y :: x \rightarrow y :: \hat{c}) \tag{33}$$

$$x \in D \leftrightarrow \mathsf{P}(x) \wedge \mathsf{catp}(x, \hat{c}) \tag{34}$$

Notice here that the membership relation between the set C and its elements requires an ontological interpretation which differs from the one required for D and its elements, and either differs from the naive view of membership. The latter would assume instantiation as its ontological counterpart ($\forall xy(x \in y \leftrightarrow x :: y)$), but then (C, D) would refer to a higher-order category. Instead, we propose to understand the set C as reflecting a category $\hat{c}$ which subsumes C's elements without C being exhaustive regarding *all* subcategories of $\hat{c}$ (hence the implication in (33)). D may either be viewed as a property $\tilde{d}$ which subsumes D's elements, or as an object category $\hat{d}$ determined by D's elements as categorial parts. Looking at formal concepts, the way in which these are ordered to form concept lattices seems to suggest the latter interpretation ($\hat{d}$). In this case, the duality of formal concepts by extent and intent can be explained by the requirement that $\hat{c}$ and $\hat{d}$ are two categories with the same extension. Furthermore, this explains the Duality Principle for Concept Lattices [1–p. 22] which states that an exchange of formal objects and attributes induces the dual concept lattice. In the ACO reading this could mean that object categories may be categorial parts of properties, or that a reinterpretation of formal objects and attributes as one type of category is needed.

Discontinuing the analysis, we need to return to the initial assumption that CPO is sufficiently expressive for every ontological interpretation of formal concept lattices. Even considering the examples in [1], this does not appear to be adequate for all of these. For instance, one example introduces countries as formal objects and organizations of countries as formal attributes. Another deals with business services as formal objects and types of business machines to which such services apply as formal attributes. Here, we admit that the ontological interpretations can be much more specialized. Moreover, the assignment to object categories and properties alone may not be suitable for all cases. Nevertheless, analyses like those above should be worthwhile for at least two reasons. Firstly, abstract core ontologies are more widely applicable, and one may refer to an existing ontological scheme when using a formalism, such as one of those above. Secondly, in the future one may expect that the small size and the applicability of abstract core ontologies leads to more extensively formalized and implemented versions than those which are available for very specialized domains. Then ACOs may be used in the very best sense of an ontology: to contribute to the interoperability of distinct formalisms.

Finally, one may ask whether an understanding of such precision is needed in applications. In our opinion, hidden imprecisions or ambiguities may cause problems or unexpected "behavior" of a formalism. For instance, the consequences of mixing the three initial interpretations I_1 (29), I_2 (30), and I_3 (31) from above remain unclear if they are not analyzed in terms of the underlying ACOs.

5 Conclusion

5.1 Summary and Discussion

Based on an analysis of meta-language issues we have provided a three-layered meta-ontological architecture which can be supported on every level by means of natural language. The architecture centers on the notion of an *abstract core ontology* (ACO), which has been discussed in detail, comprising a selection of entities and including some axiomatic fragments for two ACO variants. Roughly speaking, the CPO variant is based on objects and attributes only, whereas the CRO ontology in addition includes relations.

The last part of the paper uses CPO for the ontological foundation of basic notions of Formal Concept Analysis [1] and thus demonstrates an application of the framework apart from meta-ontological considerations. In particular, due to their limited number of selected categories ACOs are well suited for the ontological foundation of general representation formalisms, whereas rich foundational ontologies may provide too many or too special distinctions for this task. Nevertheless, the main application of the presented framework is to provide a meta-account for ontologies, including foundational ontologies.

Three important aspects of our proposal should be emphasized. Firstly, our architecture provides a clear border between ontological and formal contents. This border is situated in the transition from the meta-level to the meta-meta-level, due to which the relations between the three levels are different. In many approaches this distinction is not made, leaving it unspecified to which extent ontology plays a role on each level.

Secondly, the overall architecture is laid out with a strong interest in clarity and, in some sense, minimality in the number of notions. It is our hope that this will be beneficial in two respects. On the one hand, conceptual transparency may promote the use of the architecture in its intended areas of application. On the other hand, this shall allow for an extensive formal treatment, where experience suggests that many overlooked subtleties appear in the process of the formalization of natural language specifications. For the latter reason the formalization itself is considered valuable, as yet independent of computational applications or automated reasoning with ACOs as a primary concern.

Thirdly, during the formalization various branching points become explicit which give rise to different ACOs (with differences beyond their underlying entity types). Thus far it is not intended to promote exactly one of these ACOs. Instead, for distinct cases we expect several ACO variants to be appropriate, which can be formally related within the architecture, however.

5.2 Related Work

Two meta-level architectures shall be compared with ours. We start with the *SUO Information Flow Framework* (SUO IFF; formerly known as IFF Foundation Ontology) which appears closest in its motivation, viz. the provision of a meta-architecture for the Standard Upper Ontology (SUO) initiative:[5]

> The IFF Foundation Ontology represents metalogic. It provides a principled foundation for the metalevel (structural level) of the Standard Upper Ontology (SUO). The SUO metalevel can be used as a logical framework for manipulating collections of object level ontologies.

This quote from [29] already shows some rather strong differences between the SUO IFF and our work. First, SUO IFF pursues a meta-logical approach rather than a meta-ontological. The objects it refers to and which are manipulated on the meta-level are theories and logics, rather than single categories.[6]

SUO IFF offers a sophisticated structure which is divided into three interior levels, called lower, upper and top metalevel. It integrates a large number of mathematical concepts and appears to be based on category theory. Though this may be a powerful approach, it is fairly hard to comprehend. Overall, the SUO IFF is rather a deeply elaborated ATO than an ACO. We are not aware of a level between the SUO IFF and the proposals for SUO itself, i.e., there is no ACO level in the SUO framework.

The second meta-modeling approach to be discussed is the *Meta Object Facility* (MOF, [31]) which provides the meta-modeling architecture for UML [24]. The MOF meta-modeling architecture is derived from the "classical framework for metamodeling" [31–p. 2-2], based on four layers:

1. The *information layer* comprises data which are to be described.
2. The *model layer* describes the data which may occur in the information layer in terms of meta-data, i.e., it defines a language for some information domain. Specific UML models belong to this layer.
3. The *meta-model layer* provides descriptions of meta-data, i.e., the structure and the semantics of meta-data on the model layer. The specification of UML itself belongs to this layer.
4. The *meta-meta-model layer* describes meta-models and relates to the meta-model layer just as the meta-model layer relates to the model layer. This layer is commonly used to close by convention the regress of meta-levels.

Assuming these layers[7], the MOF approach appears fairly similar to ours, at least from an architectural point of view. In this connection it is relevant that the lowest level of our approach is concerned with ontologies, which comprise entities of both the model and the information layer of MOF. This is due to the

[5] Cf. `http://suo.ieee.org/IFF/` and `http://suo.ieee.org`, respectively.

[6] With this focus, it reminds one on other works using category theory for relating different logics, in particular the institutions of Goguen [30].

[7] Note that in MOF these layers are considered as conventions for understanding rather than being of any definitive character, i.e., the interpretation of the MOF architecture (e.g. the number of levels) may vary in dependence of the environment in which it is applied.

understanding of ontologies as specifications of conceptualizations [2], and it is also apparent from works discussing UML and ontology languages (cf. [32] and section 6 of [9]). Accordingly, the ACO level corresponds to the MOF meta-model layer, whereas the ATO level matches the MOF meta-meta-model layer.

However, there are also a number of differences compared to our approach. First, the relationships between the levels are different in the two approaches. In particular, in MOF the relations between the levels appear to be equal, namely some form of "instantiation", in contrast to the relationships between the levels as described in section 2.3. Secondly, differences originate from the object-oriented approach taken in MOF, which assumes objects with object identity, state and behavior. As far as we are aware of ontologies, in particular the dynamic aspects in the sense of software systems are not yet integrated in ontologies. A third source of differences are the details with respect to static aspects, i.e., entities and their characteristics in the formalism. For instance, MOF only allows for binary relations and has a built-in inheritance mechanism for MOF classes, whereas we have a more flexible approach regarding relations (sect. 3.1), but no inheritance. Moreover, there is a greater number of MOF entities (cf. [31–p. 3-12]), but a less formal approach is taken as regards logical formalizations. A detailed technical comparison with MOF remains to be elaborated elsewhere.

In summary, there appear to be more commonalities between our architecture and MOF than compared to the SUO IFF. Nevertheless, a closer analysis reveals rather strong differences to either of the approaches discussed, some of which seem to be justified regarding differences in scope and historical development.

5.3 Future Work

The paper concludes with indicating various ways of extending the present work. From a theoretical and philosophical point of view, further analysis of the interface between the informal and the formal levels of the architecture is needed, for which one may draw on the existing body of philosophical literature. Further, the comparison to other meta-modeling approaches as started in section 5.2 should be extended. Within the ACO level, the axiomatic fragments are to be enlarged and put into relation to each other, for which purpose the SUO IFF [33] may be used.

A major field of application which is not discussed herein are ontology languages for the Semantic Web, primarily RDF and OWL [16, 10]. For these languages one would first choose an appropriate set of ACO entities, where CRO seems to provide sufficient expressiveness[8]. However, concerning the axiomatization modifications are expected. In this connection it will be crucial to find an interpretation of the feature of RDF to allow for a self-application of RDF properties. Another important issue is the integration of datatypes. From an ontological point of view one may assume that datatypes do not require further entity types in an ACO. Rather, their treatment is expected to involve the

[8] Possibly one could even omit attributes.

denotation relation, possibly within the meta-level. Besides interpreting single features of RDF and OWL, different forms of their usage may demand different ACO interpretations. The reification of RDF properties to overcome the lack of expressiveness of these languages with respect to n-ary relations[9] is a good example in this respect.

Finally, machine-processable versions of ACOs need to be implemented. First-order theorem provers may be used to support formal investigations of established ACOs, whereas implementations, e.g. in OWL, would allow for a direct use in connection with ontologies in the Semantic Web.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
2. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5** (1993) 199–220
3. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: Wonderweb Deliverable D18: Ontology library (final). Technical report, Laboratory for Applied Ontology – ISTC-CNR, Trento (2003)
4. Heller, B., Herre, H.: Ontological categories in GOL. Axiomathes **14** (2004) 57–76
5. Pease, A., Niles, I.: IEEE Standard Upper Ontology: A progress report. Knowledge Engineering Review **17** (2002) 65–70
6. Seibt, J.: Free Process Theory: Towards a typology of processes. Axiomathes **14** (2003) 23–55
7. Sowa, J.F.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooks/Cole, Pacific Grove (2000)
8. West, M., Sullivan, J., Teijgeler, H.: ISO/FDIS 15926-2: Lifecycle integration of process plant data including oil and gas production facilities. (Data Model), ISO TC184/SC4/WG3N1328 (2003) Available from: http://www.tc184-sc4.org/wg3ndocs/wg3n1328/lifecycle_integration_schema.html.
9. DSTC, Gentleware, IBM, Sandpiper Software: Ontology Definition MetaModel: Preliminary revised submission to OMG RFP ad/2003-03-40. Technical report, Object Management Group (OMG) (2004)
10. W3C: Web Ontology Language (OWL) Specifications. W3C Recommendation, World Wide Web Consortium (W3C), Cambridge (Massachusetts) (2004) Available from: http://www.w3.org/2004/OWL/.
11. Frege, G.: Die Grundlagen der Arithmetik. Felix Meiner Verlag, Hamburg (1988) edited by C. Thiel.
12. Wittgenstein, L.: Tractatus logico-philosophicus. Routledge & Kegan Paul, London (1922) transl. by Ogden, C. K.
13. Wittgenstein, L.: Philosophische Untersuchungen. Suhrkamp, Frankfurt/Main (1967)
14. Russell, B.: On denoting. Mind **14** (1905) 479–493
15. Tarski, A.: Logic, Semantics, Metamathematics: Papers from 1923 to 1938. 2. edn. Hackett, Indianapolis (1983) ed. by Corcoran, John.

[9] Cf. http://www.w3.org/TR/swbp-n-aryRelations

16. W3C: RDF Semantics. W3C Recommendation, World Wide Web Consortium (W3C), Cambridge (Massachusetts) (2004) Available from: http://www.w3.org/TR/rdf-mt/.
17. Hayes, P., Menzel, C.: A semantics for the Knowledge Interchange Format. In: Workshop on the IEEE Standard Upper Ontology at IJCAI, Aug 6, 2001. (2001)
18. Menzel, C.: Common Logic. Available from: http://philebus.tamu.edu/cl/ (2005)
19. Devlin, K.: The Joy of Sets: Fundamentals of Contemporary Set Theory. 2. edn. Springer, Berlin (1993)
20. Barr, M., Wells, C.: Category Theory for Computing Science. 2. edn. Prentice-Hall International Series in Computer Science. Prentice-Hall, London (1995)
21. Guizzardi, G., Herre, H., Wagner, G.: On the general ontological foundations of conceptual modeling. In Spaccapietra, S., March, S.T., Kambayashi, Y., eds.: Conceptual Modeling – ER 2002, Proceedings of the 21st International Conference on Conceptual Modeling. Volume 2503 of Lecture Notes in Computer Science., Berlin, Springer (2002) 65–78
22. Pan, J.Z., Horrocks, I.: Metamodeling architecture of web ontology languages. In Cruz, I.F., Decker, S., Euzenat, J., McGuinness, D.L., eds.: Proceedings of SWWS'01, The first Semantic Web Working Symposium, Stanford University, California, USA, Jul 30 - Aug 1,2001. (2001) 131–149
23. Chaudhri, V.K., Farquhar, A., Fikes, R., Karp, P.D., Rice, J.P.: Open Knowledge Base Connectivity 2.0.3. Specification, Artificial Intelligence Center, SRI International and Knowledge Systems Laboratory, Stanford University (1998)
24. OMG: Unified Modeling Language Specification. Version 1.5, Object Management Group (OMG), Needham (Massachusetts) (2003)
25. Gracia, J.J.E.: Metaphysics and Its Tasks: The Search for the Categorial Foundation of Knowledge. SUNY series in Philosophy. State University of New York Press, Albany (1999)
26. Gracia, J.J.E.: Individuality: An Essay on the Foundations of Metaphysics. SUNY Series in Philosophy. State University of New York Press, Albany (1988)
27. Swoyer, C.: Properties. In Zalta, E.N., ed.: The Stanford Encyclopedia of Philosophy. Winter 2000 edn. Stanford University, Center for the Study of Language and Information (2000) Available from: http://plato.stanford.edu/archives/win2000/entries/properties/.
28. Loebe, F.: An analysis of roles: Towards ontology-based modelling. Onto-Med Report 7, Research Group Ontologies in Medicine, University of Leipzig (2003)
29. Kent, R.: Distributed conceptual structures. In de Swart, H.C.M., ed.: Relational Methods in Computer Science: 6th International Conference, RelMiCS 2001 and 1st Workshop of COST Action 274 TARSKI, Oisterwijk, The Netherlands, October 16-21, 2001, Revised Papers. Volume 2561 of Lecture Notes in Computer Science., Berlin, Springer (2002) 104–123
30. Goguen, J., Burstall, R.: Institutions: Abstract model theory for specification and programming. Journal of the ACM **39** (1992) 95–146
31. OMG: Meta Object Facility (MOF) Specification. Version 1.4, Object Management Group (OMG), Needham (Massachusetts) (2002)
32. Kogut, P., Cranefield, S., Hart, L., Dutra, M., Baclawski, K., Mieczyslaw, K., Smith, J.: UML for ontology development. The Knowledge Engineering Review **17** (2002) 61–64
33. Kent, R.: The IFF foundation for ontological knowledge organization. In Williamson, N.J., Beghtol, C., eds.: Knowledge Organization and Classification in International Information Retrieval. Volume 37 of Cataloging & Classification Quarterly. Haworth Press, New York (2004) 187–203