

# DOGMA-MESS: A Meaning Evolution Support System for Interorganizational Ontology Engineering

Aldo de Moor, Pieter De Leenheer, and Robert Meersman \*

VUB STARLab  
Semantics Technology and Applications Research Laboratory  
Vrije Universiteit Brussel  
Pleinlaan 2 B-1050 Brussels, Belgium  
ademoor/pdeleenh/meersman@vub.ac.be

**Abstract.** In this paper, we explore the process of interorganizational ontology engineering. Scalable ontology engineering is hard to do in interorganizational settings where there are many pre-existing organizational ontologies and rapidly changing collaborative requirements. A complex socio-technical process of ontology alignment and meaning negotiation is therefore required. In particular, we are interested in how to increase the efficiency and relevance of this process using context dependencies between ontological elements. We describe the DOGMA-MESS methodology and system for scalable, community-grounded ontology engineering. We illustrate this methodology with examples taken from a case of interorganizational competency ontology evolution in the vocational training domain.

## 1 Introduction

In collaborative communities, people sharing goals and interests work together for a prolonged period of time. For collaboration to be successful, conceptual common ground needs to be developed. Ontologies are instrumental in this process by providing formal specifications of shared semantics. Such formal semantics are a solid basis for the development of useful collaborative services and systems. However, scalable ontology engineering is hard to do in interorganizational settings where there are many pre-existing organizational ontologies and ill-defined, rapidly evolving collaborative requirements. A complex socio-technical process of ontology alignment and meaning negotiation is therefore required. Much valuable work has been done in the Semantic Web community on the formal aspects of ontology representation and reasoning. However, the socio-technical aspects of the ontology engineering process in complex and dynamic realistic settings are still little understood. A viable methodology requires not

---

\* The research described in this paper was partially sponsored by EU Leonardo da Vinci CODRIVE project B/04/B/F/PP-144.339 and the DIP EU-FP6 507483 project. The authors wish to thank Stijn Christiaens and Ruben Verlinden for their aid in the development of the methodology and system; Luk Vervenne, Roy Ackema, and Hans Wentink for their testing of the prototype system in the CODRIVE project; and Ulrik Petersen for his fast and useful updates of the Prolog+CG tool.

building a single, monolithic domain ontology by a knowledge engineer, but supporting domain experts in gradually building a sequence of increasingly complex versions of interrelated ontologies over time. Contexts are necessary to formalize and reason about the structure, interdependencies and versioning of these ontologies, thus keeping their complexity manageable. In Sect. 2, we describe our view on interorganizational ontology engineering. Sect. 3 introduces the DOGMA-MESS methodology for scalable community-grounded ontology engineering. In Sect. 4, we describe the approach to organizational ontology alignment taken in DOGMA-MESS, focusing on the relevance of organizational definitions. We end the paper with discussion and conclusions.

## 2 Interorganizational Ontology Engineering

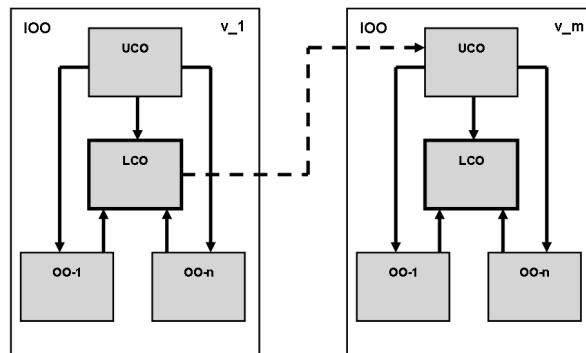
Many definitions of ontologies exist. The classical definition is that an ontology is an explicit specification of a conceptualization [12]. Other definitions, such as that an ontology is a shared and common understanding of a domain that can be communicated across people and application systems [7], stress more the community and application side of ontologies. However, the problem is not in what ontologies are, but how they *become* common formal specifications of a domain useful for building computerized services. Many open issues remain with respect to how these ontologies are to be *efficiently* engineered in communities of practice. This is all the more true in inter-organizational ontology building, where there are multiple, existing organizational ontologies that need to be aligned. In such settings, common domain ontologies need to be developed that adequately capture *relevant* interorganizational commonalities and differences in meaning. Such multiple, continuously shifting sources of meaning make knowledge sharing very difficult [8]. This is all the more true since in interorganizational settings, organizational ontologies cannot easily be merged, as they represent strong individual interests and entrenched work practices of the various participants. This means that such value-laden ontologies can only be defined in a careful and gradual process of meaning negotiation [5]. This we define as community-grounded processes for reaching the *appropriate* amount of consensus on *relevant* conceptual definitions.

Promising related work on reaching consensus on ontologies in a distributed environment has focused on architectures for consensual knowledge bases (e.g. [9]) and the cooperative construction of domain ontologies (e.g. [1]). Still, although these approaches work out basic principles for cooperative ontology engineering, they do not provide community-grounded methodologies addressing the issues of relevance and efficiency of definition processes. In interorganizational settings, however, these quality issues are of the greatest importance for ontology engineering processes to scale and be useful in daily practice. The basic question therefore is: how to develop a scalable approach to interorganizational ontology engineering? A crucial formal issue underlying such an approach is that multiple types of context dependencies need to be handled between ontological elements. Whereas much work in knowledge engineering looks at formal properties of contexts and their dependencies, in this paper we focus on how such formal approaches to handling context dependencies can be *applied* in interorganizational ontology engineering processes to increase relevance and efficiency of engineering processes. Our aim is not to be exhaustive, but to show that a systematic analysis

of such context dependencies and their use in interorganizational ontology engineering processes can help optimize this very complex socio-technical process. We focus on one very important type: specialization dependencies, which play a major role in fostering both the efficiency and relevance of interorganizational ontology engineering processes.

## 2.1 A Model of Interorganizational Ontology Engineering

We now present a generic model for understanding interorganizational knowledge engineering.



**Fig. 1.** A Model of Interorganizational Ontology Engineering

In the model, we make the following assumptions:

- An interorganizational ontology needs to be modeled not by external knowledge engineers, but by domain experts themselves. Only they have the tacit knowledge about the domain and can sufficiently assess the real impact of the conceptualizations and derived collaborative services on their organization.
- The common interest only partially overlaps with the individual organizational interests. This means that the goal is not to produce a single common ontology, but to support organizations in interpreting common conceptualizations in their own terms, and feeding back these results. A continuous alignment of common and organizational ontologies is therefore required.
- An interorganizational ontology cannot be produced in one session, but needs to evolve over time. Due to its complexity, different versions are needed.
- Starting point for each version should be the current insight about the common interest, i.e common conceptual definitions relevant for the collaborative services for which the interorganizational ontology is going to be used.
- The end result of each version should be a careful balance of this *proposal* for a common ontology with the various *individual interpretations* represented in the organizational ontologies.

Fig. 1 shows how an interorganizational ontology (IOO) consists of various, related sub-ontologies. The engineering process starts with the creation of an upper common ontology (UCO), which contains the conceptualizations and semantic constraints that are common to and accepted by a domain. Each participating organization specializes this ontology into its own Organizational Ontology (OO), thus resulting in a local interpretation of the commonly accepted knowledge. In the Lower Common Ontology (LCO), a new proposal for the next version of the IOO is produced, aligning relevant material from the UCO and various OOs. The part of the LCO that is accepted by the community then forms the legitimate UCO for the next version of the IOO.

Ontology engineering involves a number of knowledge engineering processes. Many, partially overlapping, classifications of these processes have been developed so far. Our intention with the model is not to add to these processes themselves, but to *position* them, indicating how they can be used in the bigger picture of interorganizational ontology engineering. Of course, many mappings of these processes are conceivable. In this paper, we connect only an initial, coarse-grained mapping of standard ontology engineering processes to the model. In future research, we will refine both the model and mappings of associated engineering processes.

This conceptual model of the interorganizational ontology engineering process is sufficiently specific to derive and organize practical methodological guidelines, yet generic enough to represent and compare many different approaches and techniques from an application point of view. This will help identify gaps in theory and methodologies, providing a conceptual lens to focus scattered research on a very confusing topic. In the next section, we show how this model underlies the development of STAR-Lab's own DOGMA-MESS methodology.

### 3 DOGMA-MESS

The DOGMA (Designing Ontology-Grounded Methods and Applications) approach to ontology engineering, developed at VUB STARLab, aims to satisfy real-world needs by developing a useful and scalable ontology engineering approach [17]. Its philosophy is based on a double articulation: an ontology consists of an ontology base of lexons, which holds (multiple) intuitive conceptualizations of a domain, and a layer of reified ontological commitments. These essentially are views and constraints that within a given context allow an application to commit to the selected lexons. Contexts group commitments, allowing ontological patterns to be represented and compared at various levels of granularity [2]. In this way, scalable ontological solutions for eliciting and applying complex and overlapping collaboration patterns can be built.

A fundamental DOGMA characteristic is its grounding in the linguistic representation of knowledge. This is exemplified most clearly in the linguistic nature of the lexons, with terms and role strings chosen from a given (natural) language, and that constitute the basis for all interfaces to the ontology. Linguistic "grounding" is achieved through *elicitation contexts*, which in DOGMA are just mappings from identifiers to source documents such as generalized glosses, often in natural language. As this paper however is focusing on the process architecture of interorganizational ontology building, the detailed aspects of this linguistic grounding fall mostly outside of our scope.

### 3.1 Outline of the Methodology

The efficiency and relevance of eliciting and applying ontological knowledge is at the heart of the DOGMA methodology. However, still undeveloped was the layer in which the DOGMA ontology engineering processes are grounded in communities of use. This is the purpose of the DOGMA-MESS methodology.

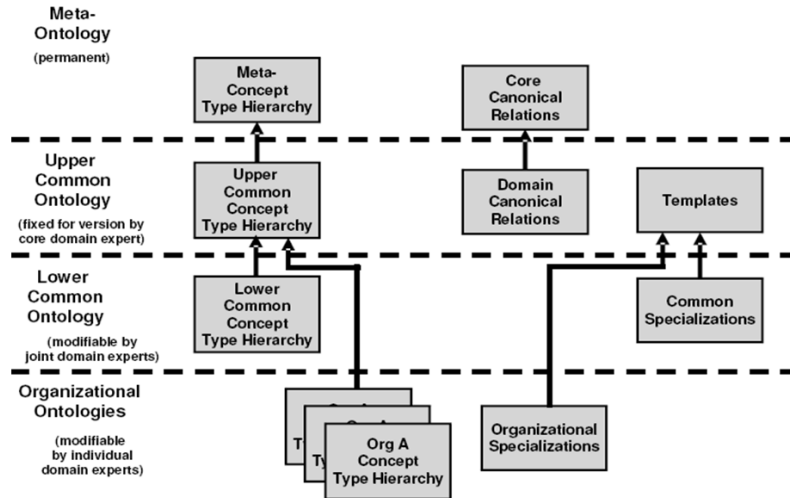


Fig. 2. Interorganizational Ontology Engineering in DOGMA-MESS

Based on our model of interorganizational ontology engineering, the approach adopted in DOGMA-MESS is characterized in Fig. 2. Arrows in the diagram indicate specialization dependencies between ontologies. Each version of the IOO construction consists of three stages: (1) creation of the templates; (2) definition of the organizational specializations (divergence of definitions); and (3) definition of the common specializations (convergence of definitions). After that, the relevant knowledge to be retained is moved to the first stage of the next cycle (still under design). Some important properties of the intra-version processes are:

- A (permanent) *Meta-Ontology* is the same for all applications of DOGMA-MESS and only contains stable concept types like ‘Actor’, ‘Object’, ‘Process’, and ‘Quality’. Three main types of Actors are defined: *Core Domain Experts* represent the common interest, *Domain Experts* represent the various organizational interests, and *Knowledge Engineers* help the other experts define and analyze the various ontologies. The Meta-Ontology also contains a set of core canonical relations, similar to the ones described in [16], such as the ‘Agent’, ‘Object’, and ‘Result’-relations.
- Each domain has its own *Upper Common Ontology*, and is maintained by the core domain expert. It first of all contains a specialization of the concept type hierarchy of the Meta-Ontology. This *Upper Common Concept Type Hierarchy* organizes

the (evolving) concept types common to the domain. Domain canonical relations specialize core canonical relations in terms of the domain. For instance, whereas ‘Agent’ is a core canonical relation, in a particular domain this may be translated into ‘Person’. The most important type of construct in the UCO are the *Templates*. A template describes a common knowledge definition most relevant to the common interest. At the beginning of each new version, the core domain expert defines templates that best capture the focus interests of that moment. Over time, templates should become more numerous and specialized.

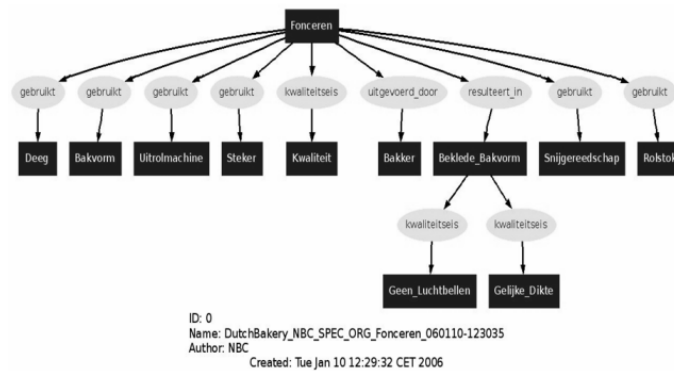


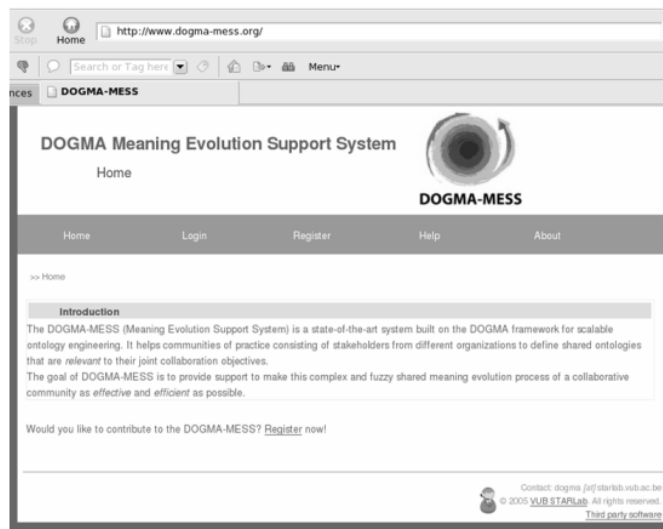
Fig. 3. An organizational specialization of a task template

- Templates are specialized into *Organizational Specializations* by the domain experts representing the various organizations. To this purpose, domain experts can add concept types that are particular to their organization. These *Organizational Concept Type Hierarchies* themselves need to be a specialization of the Upper Common Concept Type Hierarchy. Fig. 3 gives an example from the CODRIVE project (see below) of a definition by one of the partners in the Dutch Bakery case having specialized the task template for the task ‘Panning’ (= ‘Fonceren’ in Dutch).
- The most important layer for meaning negotiation is the *Lower Common Ontology*. This is where the ‘specification agenda’ as represented by the UCO and the, often widely differing, organizational interpretations need to be aligned and the most relevant conceptualizations for the next version need to be selected. This process is far from trivial. In the current implementation of DOGMA-MESS, there is only a very simple rule: all (selected) definitions need to be full specializations of the templates, hence they are called Common Specializations. Likewise, the Lower Common Concept Type Hierarchy needs to be a specialization of the Upper Common Concept Type Hierarchy. This, however, is overly simplified. In the meaning negotiation process, new definitions may be created that are not (complete) specializations, but represent a new category of template for the next version of the IOO, for example. This is where many of the DOGMA existing ontology analysis processes, for example based on lexon and commitment comparison, as well as ORM constraint

analysis may play a role. At any rate, our framework allows for such methodology evolution to be clearly described. In Sect. 4.3, we give an example of one of the LCO processes currently being developed: organizational ontology alignment.

### 3.2 System Implementation

The system supporting the DOGMA-MESS methodology is being implemented as a web server that can be accessed by any web browser, thus ensuring maximum accessibility and ease-of-use (Fig. 4).



**Fig. 4.** The DOGMA-MESS system

The core of the server is a Java server that interacts with the DOGMA Studio server, which in turn implements the standard DOGMA ontology engineering and analysis engine. Special converters translate standard DOGMA representations to and from a simple native DOGMA-CG format (amongst other things limited to simple referents). CharGer is one of the tools allowing core domain experts to create templates in CharGer-XML, which can also be converted into DOGMA-CG. Concept type hierarchies can be imported as indented text files, easily to be edited with simple ASCII-editors. Type hierarchies and templates, like organizational specializations, can also be edited through the DOGMA-MESS web interface. This interface, for example, allows concept and relation types to be defined and graph concepts and relations to be added, changed, or removed. Prolog+CG has been embedded in DOGMA-MESS as the conceptual graph inference engine. The main operation currently being used is the projection operation for checking whether organizational specializations conform to their templates. To visualize stored and inferred graphs, AT&T's GraphViz engine is being used.

### 3.3 Initial User Experiences

The explicit mission of DOGMA-MESS is to increase the efficiency and relevance of the interorganizational ontology engineering process. A continuous process of testbed-like development of methodology and system is therefore essential. Its socio-technical fine-tuning is being done in the CODRIVE project. This project aims to develop a methodology for shared competency ontology definition between organizations representing the educational sector, industry, and public employment agencies. Common competency models are essential for the interoperability of the information systems of these organizations. This in turn is required for them to provide better training and job matching services, which by their very nature require close interorganizational collaboration.

A first full version of DOGMA-MESS is operational now. We are currently experimenting with experts representing multiple educational organizations in the Dutch bakery sector. Pilot tests have been done, and a set of templates is currently being specialized by a number of domain experts representing various Dutch Bakery schools, under supervision of a core domain expert from the Dutch National Bakery Centre. Initial user experiences with tool and system are generally positive. The most serious limitation is complexity rather than lack of functionalities. One of our main development activities therefore is experimentally simplifying workflows and interfaces. For example, initially users had to define their concept types before they were going to create a definition. However, often they find out which concept type they need only *during* definition construction. We now allow for a type creation process to be spawned from a definition creation process, and afterwards to return to the right step in that originating process.

After a face-to-face demo and one hour practice session, most are able to create specializations online asynchronously. Task and competency templates have been defined for the hundreds of tasks that bakery students need to master by the core domain expert. A task template, for example, has the task as the *focus concept*, around which relations need to be specialized such as who is the person doing the task, what resources, materials, equipment, and tools are needed as inputs, what is the resulting output, and what quality aspects are required for task components.

The templates are currently being specialized by these experts into organizational specializations (see Fig. 3 for an example of such a specialization). With most start-up problems having been addressed, the initial knowledge base of dozens of definitions is now growing into hundreds of definitions. Ultimately, thousands, tens of thousands, or even more definitions will be obtained in a typical domain. Given that time of these experts is very expensive, they should only focus on the most relevant definitions at any moment in time. How to reach this definition convergence after the divergence of the organizational specialization process is the aim of the third stage of each ontology version. Ontology alignment is a key issue here.

## 4 Organizational Ontology Alignment in DOGMA-MESS

Ontology alignment means making -literally- common sense out of a set of individual ontologies. This is the most difficult ontology engineering process, with the most of degrees of freedom for operationalization. Formalization of the ontology engineering process is required to ensure the quality of the design of this process.

### 4.1 Characterizing Ontology Engineering Processes

An important class of ontology engineering processes concerns *ontology integration*. This process has been studied extensively in the literature (for a state-of-the-art survey, cf. [10, 14]). Although different groups vary in their exact definition, ontology integration is generally considered to consist of four key subprocesses: the *articulation*, *mapping*, *alignment*, and *merging* of ontologies. Ontology articulation deals with the problem of how to define conceptualizations. Ontology mapping concerns how to link elements from different ontologies, alignment how meanings common to different ontologies can be identified, and merging focuses on how to generate completely new ontologies out of existing ones. Interorganizational ontology engineering to a large extent focuses on the first three subprocesses, merging being of less relevance.

All ontology integration methodologies use some combination of these *macro-ontology engineering processes*. However, in their operational implementation of these processes, which we call *micro-ontology engineering processes*, methodologies differ widely. We use the following (non-exhaustive) set of micro-process primitives: the creation, modification, termination, linking, and selection of ontological definitions. The *creation* of a definition generates a new definition, often from a template. A *modification* changes concepts and relations in an existing definition. A *termination* removes the definition from the ontology. These operations were described in detail in terms of pre and post conditions in [4]. Still lacking were formal definitions of the micro-processes of explicitly *linking* ontologies and elements from ontologies and *selecting* relevant definitions from a (growing) ontology. Many operationalizations of these processes have already been developed in the field. The question is how to apply and (re)combine them to increase the quality of real-world ontology engineering processes.

*Contexts* are important building blocks in our decomposition and linking of ontology engineering processes [2]. We define a context of an entity as the set of circumstances surrounding it. *Context dependencies* constrain the possible relations between the entity and its context. Many different types of context dependencies exist, within and between ontological elements of various levels of granularity, ranging from individual concepts of definitions to full ontologies. One of the best studied dependencies, which we focus on in this paper, are specialization dependencies. For instance, an organizational definition of a particular task (the entity) can have a specialization dependency with a task template (its context). The constraint in this case is that each organizational definition must be a specialization of the template. In conceptual graphs terms, this would mean that the template must have a projection into the organizational definition. We give an exhaustive analysis of such dependencies in interorganizational ontology engineering in [3]. In Sect. 4.3, we will only illustrate specialization dependencies,

by formally describing and decomposing one type of ontology integration (macro)-process: the alignment of organizational ontologies.

## 4.2 A Formalization of Specialization Dependencies

We formalize the DOGMA-MESS methodology in terms of a set of ontologies and their (specialization) context dependencies. First, we define an ontology as a logical theory:

**Ontology** An ontology is defined as a structure  $O = \langle S, A \rangle$ , where  $S$  is the signature and  $A$  is a set of ontological axioms. The signature typically consists of a set of concept symbols and relation symbols, the latter denotes relations whose arguments are defined over the concepts. The axiomatization specifies the intended interpretation of the signature. It essentially defines which relation symbol  $r$  in  $S$  is to be interpreted as subsumption relation. Formally, this requires that  $r$  defines a strict partial order (poset). Furthermore,  $A$  optionally defines a strict partial order on the relation symbols in  $S$  (thus defining a specialization hierarchy on the relations), and a particular sort or class of axioms (or semantic constraints), depending on the kind of ontology.

**Specialization Dependencies** These dependencies, illustrated by arrows in Fig. 2, are an important context dependency in DOGMA-MESS, and are used to connect the various interorganizational ontology entities. Conceptual graph theory is one of the most universal and powerful formalisms for dealing with specializations, with its concept and relation type hierarchies and generalization hierarchies of graphs. We use it here to further define and operationalize our methodology. A conceptual graph can be represented by a logical theory  $\langle S, A \rangle$ , where the signature consists of concept types, canonical relation types, and a set of ontological definitions (CGs). The axiomatization consists of a concept type hierarchy and a relation type hierarchy defined by partial orders in terms of concept types and relation types in  $S$  respectively. Most ontological entities, such as the definitions stored in an ontology (i.e. template, organizational specializations), can be represented as conceptual graphs, and the usual conceptual graph operations can be applied to reason about them.

- An interorganizational ontology  $IOO$  contains the following sub-ontologies: a meta-ontology  $MO$ , an upper common ontology  $UCO$ , a lower common ontology  $LCO$ , and a set of organizational ontologies  $OO_i$ , one for each member organization.
- Each ontology  $O$  contained in  $IOO$  consists of a concept type hierarchy  $CTH$  and a set of ontological definitions  $D$ . Each definition  $d \in D$  is a well-formed conceptual graph.
- The meta-ontology  $MO$  consists of a meta-concept type hierarchy  $CTH_M$ , and an optional set of core canonical relations  $CR_M$ . Each relation  $cr_m = \langle c_1, r, c_2 \rangle \in CR_M$ , with  $c_1, c_2 \in CTH_M$ , and  $r \in CR$ , which is a standard set of relation types similar to the canonical relation types described in [16].
- The upper common ontology  $UCO$  consists of a upper common concept type hierarchy  $CTH_{UC}$ , an optional set of domain canonical relations  $CR_{UC}$ , and a non-empty set of templates  $T$ .  $CTH_{UC} \leq CTH_M$  and  $CR_{UC} \leq CR_M$ , in the standard CG theory sense.
- The lower common ontology  $LCO$  consists of a lower common concept type hierarchy  $CTH_{LC}$ , and a set of common specializations  $D_{LC}$ . At the start of a version period  $D_{LC} = \emptyset$ .  $CTH_{LC} \leq CTH_{UC}$  and  $\forall d_{lc} \in D_{LC} : \exists t \in T \wedge d_{lc} \leq t$ .

- Each organizational ontology  $OO_i$  consists of an organizational concept type hierarchy  $CTH_{O_i}$  and a set  $D_{O_i}$  of organizational specializations of templates from the UCO, with  $CTH_{O_i} \leq CTH_{UC}$  and  $\forall d_{O_i} \in D_{O_i} : \exists t \in T \wedge d_{O_i} \leq t$ .
- Each of the constructs defined above is indexed by a version number  $v$ . For clarity, this index number is not shown in the definitions.

### 4.3 Selecting Relevant Organizational Specializations

In this section, we formalize the process of aligning organizational ontologies in DOGMA-MESS by selecting the most relevant organizational specializations as the common specializations (see Fig. 2). Such a process increases the *relevance* of definitions, since the community will focus on those definitions most in line with its (evolving) goals. The rationale is that templates at first will be coarse, as the community is still learning about its goals and interests of its members. Over time (and versions), however, templates can become more focused. This requires that only the most relevant definitions are passed on as templates (and possibly other domain definitions) to the UCO of the next version, since time and energy of domain experts are limited. The notion of relevance of ontological definitions in an evolving collaborative community is still almost unexplored in the ontology engineering literature.

As collaborative communities evolve and learn, their number of ontological definitions, often specializations of earlier definitions, grows. Furthermore, the collaborative goals and requirements become clearer over time, often leading to new types of definitions that need to be created. At the same time, certain older ones become less relevant because of the shifting collaborative focus. The process of creating and modifying ontological definitions is very expensive, since many experts are to be involved, who often have to consult their respective organizations before being able to make a commitment.

To increase relevance and efficiency of the interorganizational ontology engineering process, some way of selecting the most relevant definitions in a particular stage of the evolution of the interorganizational ontology is needed. DOGMA-MESS has currently implemented this selection process in the following way:

- The organizational ontology alignment process starts the moment the set of organizational ontologies  $OO_i$  has been updated for the current version  $v$  of the  $IOO$ .
- The community defines a set of *relevance definitions*  $D_R$ .

**Example:** a group of bakers started of with making definitions of general baking tasks, but has now discovered that the real education gap is in baking sweet products:

$$D_R = \{ \text{Baking} \rightarrow \text{produces} \rightarrow \text{Sweet Stuff} \}$$

- Each  $d_r \in D_R$  is now *lexonized*, which means that it is automatically flattened into a set of lexons  $L_R$ . Lexons are similar to binary conceptual relations, with a role/co-role pair instead of a single connecting relation type. If no role/co-role mapping exists in the DOGMA knowledge base, the co-role is left empty. The co-role helps to find additional linguistic forms of the same conceptual relation. Using these lexons, the set of *relevance relations*  $R_R$  now is formed by all "surface forms" of relevant conceptual relations (i.e. creating a conceptual relation from a role, and

another relation from the co-role, arrows inverted).

**Example:**  $R_R = \{ \text{Baking} \xrightarrow{\text{produces}} \text{Sweet\_Stuff}, \text{Sweet\_Stuff} \xrightarrow{\text{is\_produced\_in}} \text{Baking} \}$

- For each organizational specialization  $d_{O_i}$  in each organizational ontology  $OO_i$ , a relevance score  $s_r$  is now computed, by checking if the relevance relations project into the definition.  $\forall d_{O_i} \in D_{O_i}$ , with  $CTH_{O_i}$  :
  - $s_r(d_{O_i}) = 0$ .
  - $\forall r_r \in R_R$  : if  $\exists \pi r_r$  in  $d_{O_i}$ , then  $s_r(d_{O_i}) = s_r(d_{O_i}) + 1$ .

**Example:** assume Baker A is an expert in cakes, and always stresses that cakes should be just sweet enough. His organizational ontology  $OO_A$  therefore contains this organizational task specialization  $d_{O_A}$ :



Also, in his concept type hierarchy  $CTH_{O_A}$ ,  $\text{Cake} < \text{Sweet\_Stuff}$ . Since the second relevance relation projects into this graph, its relevance score is increased by one.

- Now, all definitions have an associated relevance score. The scores can now be used to rank the relevance of the organizational definitions to the common interest. Once ranked, either the  $x$  highest ranked definitions can be selected, or those definitions that meet a certain threshold. If the community is very busy, or has only very limited time, the threshold level can be set higher, so that less definitions need to be considered for lifting into the  $UCO$  of  $IOO$  version  $v + 1$ .

Of course, this is only a crude measure of relevance. Important is that a start has been made with operationalizing this fuzzy, but necessary idea and grounding it in a useful interorganizational ontology engineering methodology, from where it can evolve into more sophisticated approaches. The relevance assessment procedure is currently being experimented with in the CODRIVE project as a way of increasing focus and thus motivation of users.

## 5 Discussion

This paper has made several contributions. A conceptual model of interorganizational ontology engineering was presented, and the beginning of its formalization, which can be used to analyze and compare ontology engineering methodologies. DOGMA-MESS was introduced, as a prime example of an interorganizational ontology engineering methodology. One core IOE process, organizational ontology alignment, was described. This process explicitly addresses the fundamental problem of keeping ontological definition processes relevant, a necessary condition for these complex socio-technical processes to scale.

The aim of this paper was not to examine any particular ontology engineering process in depth, much work has been done in this respect in, for example, Semantic Web research. Neither was it to provide a full theory of ontological context dependency management. Substantial work already exists on this in the Conceptual Structures community, e.g. [16, 15, 11]. Also outside this community much relevant work already exists. For example, Guha et al. [13] adopt a notion of context primarily for scaling the

management of the very large knowledge base Cyc. Our aim, however, was to introduce a framework and concrete methodology to apply this valuable theoretical work to a very pressing organizational and societal problem: making interorganizational ontology engineering work. In other words, our interest is how to apply such semantic techniques to community meaning negotiation goals [6].

In future work, we will refine our conceptual model of interorganizational ontology engineering and position related methodologies in our descriptive framework. Extensive intra and cross-case analyses will be done, providing both valuable data and extensions and refinements of the DOGMA-MESS methodology and system.

Having specialization and other context dependencies clearly defined and inform the design of knowledge definition processes is a necessary but not a sufficient condition for guaranteeing efficient and relevant interorganizational ontology engineering processes. A thorough grounding in socio-technical principles of community informatics, such as legitimate meaning negotiation, is required for interorganizational ontology engineering to succeed [5]. Systematically supporting this meaning negotiation process through a community portal and guided discussion facilities, tightly linked to the DOGMA-MESS knowledge base, is one of our key research foci at the moment.

## 6 Conclusions

In this paper, literally speaking, we have put ontology engineering in context. We have characterized interorganizational ontology engineering as a process involving different domain experts, with well-defined roles, producing increasingly specialized versions of composite interorganizational ontologies.

Our goal is to efficiently produce relevant ontological definitions. To support this complex process, we introduced our DOGMA-MESS methodology. It combines a comprehensive socio-technical methodology with a formal framework of specialization context dependencies. With it, macro-processes like ontology alignment can be decomposed into manageable combinations of micro-processes, such as template creation and specialization. Moreover, various ways of relevance scoring can support the subtle socio-technical dynamics in expert communities of practice. There is thus not one right way of designing these macro/micro processes. We aim to expand our work in intensive empirical evaluations of DOGMA-MESS in and across a range of high-impact cases.

Our aim was not to produce a comprehensive formal definition of macro and micro ontology engineering processes nor of context dependencies. This would have been far too ambitious, nor do justice to the large amount of related work. Rather, our methodology is in fact a meta-methodology that can accommodate many different ontological methodologies and technologies, synthesizing and tailoring them to the needs of real communities of use.

By positioning related work using our conceptualization of interorganizational ontology engineering and its practical implementation in a working methodology and system, the factors influencing the relevance and efficiency of this extremely complex socio-technical process can be better understood. This understanding should considerably progress the identification of research gaps, alignment of research efforts, and applicability of results from ontology integration research. In the end, ontologies are not

a goal in themselves, but instruments to facilitate collaborative *community dynamics*. We are confident that DOGMA-MESS will help improve this dynamics by discovering new ways for communities to find conceptual common ground.

## References

1. F.-R. Aschoff, Schmalhofer, F., and L. van Elst. Knowledge mediation: A procedure for the cooperative construction of domain ontologies. In *Proc. of the ECAI 2004 Workshop on Agent-Mediated Knowledge Management*, pages 29–38, 2004.
2. P. De Leenheer and A. de Moor. Context-driven disambiguation in ontology elicitation. In P. Shvaiko and J. Euzenat, editors, *Context and Ontologies: Theory, Practice, and Applications. Proc. of the 1st Context and Ontologies Workshop, AAAI/IAAI 2005, Pittsburgh, USA, July 9, 2005*, pages 17–24, 2005.
3. P. De Leenheer, A. de Moor, and R. Meersman. Context dependency management in ontology engineering. Technical Report STAR-2006-03-01, VUB STARLab, Brussel, March 2006.
4. A. de Moor. *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*. PhD thesis, Tilburg University, The Netherlands, 1999. ISBN 90-5668-055-2.
5. A. de Moor. Ontology-guided meaning negotiation in communities of practice. In P. Mambrey and W. Gräther, editors, *Proc. of the Workshop on the Design for Large-Scale Digital Communities at the 2nd International Conference on Communities and Technologies (C&T 2005), Milano, Italy, June 2005*, 2005.
6. A. de Moor. Patterns for the pragmatic web. In *Proc. of the 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17-22, 2005*, pages 1–18, 2005.
7. S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, S. Melnik, M. Klein, and J. Broekstra. Knowledge representation on the Web. In *Proc. of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany, 2000*.
8. T. Edgington, B. Choi, K. Henson, T.S. Raghuram, and A. Vinze. Adopting ontology to facilitate knowledge sharing. *Communications of the ACM*, 47(11):217–222, 2004.
9. J. Euzenat. Building consensual knowledge bases: Context and architecture. In N.J.I. Mars, editor, *Towards Very Large Knowledge Bases - Proceedings of the KB&KS '95 Conference*, pages 143–155. IOS Press, 1995.
10. J. Euzenat, T. Le Bach, J. Barrasa, et al. State of the art on ontology alignment. Knowledge Web Deliverable KWEB/2004/d2.2.3/v1.2, 2004.
11. B. Ganter and G. Stumme. Creation and merging of ontology top-levels. In *Proc. of the 11th International Conference on Conceptual Structures, ICCS 2003 Dresden, Germany, July 21-25, 2003*, pages 131–145, 2003.
12. T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
13. R. Guha and D. Lenat. Cyc: a midterm report. *AI Magazine*, 11(3):32–59, 1990.
14. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: The state of the art. In *Proc. of the Dagstuhl Seminar on Semantic Interoperability and Integration (Dagstuhl, Germany), 2005*.
15. G. Mineau and O. Gerbé. Contexts: A formal definition of worlds of assertions. In *Proc. of the 5th International Conference on Conceptual Structures, ICCS '97, Seattle, Washington, USA, August 3-8, 1997*, pages 80–94, 1997.
16. J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.
17. P. Spyns, R. Meersman, and M. Jarrar. Data modelling versus ontology engineering. *SIGMOD Record*, 31(4):12–17, 1998.