Context-driven Disambiguation in Ontology Elicitation *

Pieter De Leenheer and **Aldo de Moor**
Semantics Technology and Applications Research Laboratory (STARLab)
Department of Computer Science
Vrije Universiteit Brussel
Pleinlaan 2, B-1050 BRUSSEL, Belgium
{pieter.de.leenheer,aldo.de.moor}@vub.ac.be

Abstract

Ontologies represent rich semantics in a lexical way. Lexical labels are used to identify concepts and relationships, though there is no bijective mapping between them. Phenomenons such as synonyms and homonyms exemplify this, and can result in frustrating misunderstanding and ambiguity. In the elicitation and application of ontologies, the meaning of the ontological knowledge is dependent on the context. We consider the role of context in ontology elicitation by introducing context in a concept definition server for ontology representation. We also adopt other features of context found in literature, such as packaging of knowledge, aligning elements of different contexts, and reasoning about contexts. Finally, we illustrate context-driven ontology elicitation with a real world case study.

Introduction

Though a vast amount of research has been conducted on formalising and applying knowledge representation (KR) models (Gruber 1993; Guarino 1998; Meersman 1999; Ushold & Gruninger 1996; Farquhar, Fikes, & Rice 1997), there is still a major problem with lexical disambiguation and subjectivity during the *elicitation* and *application* of an ontology. The problem is principally caused by two facts: (i) no matter how expressive ontologies might be, they are all in fact lexical representations of concepts, relationships, and semantic constraints; and (ii) linguistically, there is no bijective mapping between a concept and its lexical representation.

During the *elicitation* of an ontology (cfr. Fig. 1), its basic knowledge elements (such as concepts and relationships) are extracted from various resources such as (structured) documents and human domain experts. Many ontology approaches focus on the conceptual modelling task, hence the distinction between lexical level (term for a concept) and conceptual level (the concept itself) is often weak or ignored. In order to represent concepts and relationships lexically, they usually are given a uniquely identifying term (or label). However, the context of the resource the ontology element was extracted from is not unimportant, as the meaning of a concept behind a lexical term is influenced by this *context of elicitation*. Phenomenons such as synonyms and homonyms are typical examples of this, and can result in frustrating misunderstanding and ambiguity when unifying information from multiple sources. Similar for the *application* of an ontology: the interpretation of the knowledge elements (which are referred to by terms) of the ontology is ambiguous if the context of application, such as the purpose of the user, is not considered.

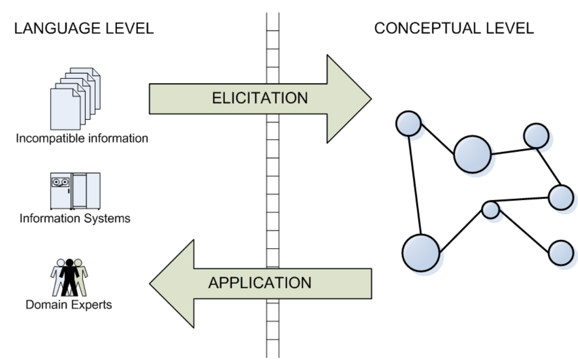

Figure 1: Ontologies are elicited by extracting knowledge from various sources and are also applied in different contexts.

Context was already introduced before to tackle lexical disambiguation (Buvač 1996b; Meersman 2001). In the literature, different, often unrelated interpretations of context in KR can be found. E.g., researchers introduced context to provide subjectivity: a context is a grouping of knowledge that provides a subjective (i.e. "context-dependent") view of a particular (community of) agent(s) (Guha & D. 1990; Lenat 1995; Theodorakis 1999). Our purpose in this paper, however, is to examine in detail the role that multiple contexts can play in the disambiguation of terms in the ontology elicitation process.

This paper is structured as follows: first, we give a synthesis of our literature study on context, and identify the fea-

*We would like to thank our colleagues in Brussels, especially Robert Meersman and Luk Vervenne, for the valuable discussions about theory and case. This research has been partially funded by the EU Leonardo da Vinci Project CODRIVE (BE/04/B/F/PP-144.339)

tures of contexts useful for our purpose. Then we present the DOGMA ontology representation framework, where we introduce the idea of context and a concept definition server to (i) logically group parts of knowledge, (ii) disambiguate the lexical representation of concepts and relationships, by distinguishing between language level and conceptual level, and (iii) build semantic bridges between different ontological contexts. We illustrate our framework by considering context-driven ontology elicitation in a real-world case study.

Contexts, Situations and Possible Worlds

Today in AI and linguistics, the word *context* has gained a (confusing) variety of meanings, which have led to diverse interpretations and purposes of context (Sowa 1995; 1997). Moreover, context is found in various AI application fields such as database integration (Farquhar *et al.* 1995), knowledge translation (Buvač & Fikes 1995), and reasoning (Giunchiglia 1993; Nayak 1994; McCarthy & Buvač 1994).

Peirce (Buchler 1955), with his preliminary notion of *sheets of assertion* was one of the pioneers in the formalisation of context. Although well known, there is no common understanding of its semantics. The theories of semantics based on *possible worlds* (Kripke 1963) or *multi-modal logic* are also associated with the notion of context. A proposition is assigned a true or false value depending on which world (read: "context") is considered among a set of possible worlds. Amongst this set of possible worlds there is the actual world, assigned w_0. Some worlds are accessible from some other worlds. A proposition is *necessarily* true in w_0 if it is true in every world accessible from w_0. A proposition is *possibly* true if it is true in some (at least one) possible world accessible from w_0. Instead of assuming possible worlds, Hintikka (1963) developed independently an equivalent semantics for multi-modal logic, which he called model sets.

McCarthy (1987; 1993) formalised context as first-class objects by introducing his basic relation $ist(\mathcal{C}, p)$. This ist predicate might be read as: "proposition p is true in context $\mathcal{C}$".

Situation semantics (Barwise & Perry 1983) is a reaction to multi-modal logic. While each possible worlds' model represents an large, open-ended, unbounded region of spacetime, *situations* are smaller chunks that are more "manageable" (Sowa 2000, pp. 184).

Guha et al. (1990), adopt a notion of context for scaling the management of the very large knowledge base Cyc (Lenat 1995). They implement *microtheories* that allow assumptions in a limited context, but leave open the ability to use the knowledge in a larger context. Microtheories are organised in an subsumption hierarchy, where everything asserted in the super-microtheory, is true in the sub-microtheory, unless explicitly contradicted. Examples are theories of bibliography keeping, theories of car selling company, etc. Similar to McCarthy's conception, microtheories are inter-related via lifting rules stated in outer context.

McCarthy's intention to use context was predominantly for reasoning about relationships *between* contexts in an *outer context*. A proposition that is true in one context, might be asserted in another context under certain conditions. He calls this *lifting*. For example the same predicate p in context $\mathcal{C}_1$ can have a different name or arity, or be equivalent in another context $\mathcal{C}_2$. Generally, a lifting formula specifies an alignment between proposition and terms in subcontexts to possibly more general propositions and terms in an outer context. Subcontexts are often specialised with regard to time, place, and vocabulary. According to McCarthy however, there is no "root" context in which all axioms hold and everything is meaningful: for each axiom in a given context, one can always find a more general context in which the axiom fails.

McCarthy's work was further developed by Buvač and Guha (1991). Buvač (1996b) concluded that there is a need for common sense knowledge to lexically disambiguate a proposition like "Vanja is at a bank". From this proposition, the inference system cannot determine whether Vanja is at a river bank or at a financial bank. Note that McCarthy (1993) also concluded this as one of the reasons for introducing his formal context. He argued similarly that a term might have a particular meaning in a professional conversation different from the one in daily language use. This trend is reinforced by the field of linguistics (Langacker 1987).

Buvač (1996a) also extended the propositional logic of context to the quantificational case, providing a way to express first-order properties of contexts, and the truth of arbitrary predicates within a context.

Theodorakis (1999) introduces a context formalism from the perspective of data modeling. A context is an abstract object containing other objects, viz. data models. He organises different data models constructed from different perspectives or levels of abstraction into different contexts. These contexts are "integrated" by cross-referencing one to another. This is very similar to Guha's micro-theories.

In conceptual graph theory (Sowa 2000), context provides a means to describe what is true in a certain situation, without requiring the description of these situations per se (Mineau, Missaoui, & Godinx 2000). Mineau et al. (1997) propose to structure the world into a partial order of subsumption between contexts. The most general context (which does not exist in McCarthy's opinion) is called, in honour of Peirce, the *universal sheet of assertion*.

Giunchiglia (1993) was especially motivated by the problem of reasoning on a subset of the global knowledge base. The notion of context is used for this "localisation". His perspective is similar to (McCarthy 1987; 1993).

Guha (1991) defined a mapping from McCarthy's contexts to *situations* of Barwise and Perry (1983), i.e., for every context $\mathcal{C}$, there exists a situation s *and* for every proposition p: p is true in context $\mathcal{C}$ if and only if situation s is described by p. A similar parallel exists between logic of context and standard multi-modal logic. For more we refer to Halpern and Moses (1992).

Synthesis Based on our literature study, we distinguish four effective features of context (for reasoning), which we aim to integrate in our framework: (i) contexts package related knowledge: in that case a context defines (part of) the knowledge of a particular domain; (ii) context provides pointer for lexical disambiguation; (iii) lifting rules provide an alignment between assertions in disconnected knowledge bases (or contexts); and (iv) "Statements about contexts are themselves in contexts" (McCarthy 1996); in other words, contexts can be embedded.

Summarising, to disambiguate terms, in general, an analysis of multiple contexts is needed. However, to implement these theoretical notions in real world systems is not trivial. Such implementation is the focus of the DOGMA ontology framework.

DOGMA Ontology Framework

DOGMA[1] is an ontology representation model and framework that separates the specification of the *conceptualisation* (i.e. lexical representation of concepts and their interrelationships) from its *axiomatisation* (i.e. semantic constraints). This principle corresponds to an orthodox *model-theoretic* approach to ontology representation and development. Consequently, the DOGMA framework consists of two layers: a *Lexon Base* and a *Commitment Layer*. A full formalisation of DOGMA Ontology is found in (De Leenheer & Meersman 2005).

Lexon Base

The Lexon Base is an uninterpreted, extensive and reusable pool of elementary building blocks for constructing an ontology. These building blocks (called *lexons*[2]) represent *plausible binary fact-types* (e.g., Person drives/is_driven_by Car). The Lexon Base is stored in an on-line DOGMA server. For guiding the ontology engineer through this very large database, *contexts* impose a meaningful grouping of these *lexons* within the Lexon Base. The context of a lexon refers to the source it was extracted from. Sources could be terminological[3] or human domain experts. A lexon is defined as:

Definition 1 *A lexon is an ordered 5-tuple of the form* $< \gamma, t_1, r_1, r_2, t_2 >$ *where* $\gamma \in \Gamma$, $t_1 \in T$, $t_2 \in T$, $r_1 \in R$ *and* $r_2 \in R$. Γ *is a set of identifiers,* T *and* R *are sets of strings in some alphabet* $\mathcal{A}$*;* t_1 *is called the headword of the lexon and* t_2 *is called the tailword of the lexon;* r_1 *is the role of the lexon,* r_2 *is the co-role;* γ *is the context in which the lexon holds.*

Role and co-role indicate that a lexon can be read in two directions. A lexon $< \gamma, t_1, r_1, r_2, t_2 >$ is a fact type that might hold in a domain, expressing that within the context γ, an object of type t_1 might plausibly play the role r_1 in relation to an object of type t_2. On the other hand, the same lexon states that within the same context γ, an object of type t_2 might play the co-role r_2 in (the same) relation to an object of type t_1.

[1]acronym for Developing Ontology-Guided Mediation of Agents; a research initiative of VUB STARLab

[2]Lexons are DOGMA knowledge elements.

[3]"A context refers to text, information in the text, to the thing the information is about, or the possible uses of the text, the information in it or the thing itself" (Sowa 2000, pp. 178).

Some role/co-role label pairs of lexons in the Lexon Base might intuitively express a specialisation relationship, e.g. $< \gamma, manager, is\,a, subsumes, person >$. However, as already mentioned above: the lexon base is uninterpreted, so the decision to interpret a role/co-role label pair as being a part-of or specialisation relation, is postponed to the commitment layer, where the semantic axiomatisation takes place.

Commitment Layer

Committing to the Lexon Base means selecting a meaningful set Σ of lexons from the Lexon Base that approximates well the intended[4] conceptualisation, and subsequently putting semantic constraints on this subset. The result (i.e., Σ plus a set of constraints), called an *ontological commitment*, is a logical theory that intends to model the meaning of this application domain. An ontological commitment constitutes an axiomatisation in terms of a network of lexons logically connected and provides a partial view of the Lexon Base. These networks are visualised in a NIAM[5]-like schema (cfr. Fig. 2). An important difference with the underlying Lexon Base is that commitments are internally unambiguous and semantically consistent[6]. Once elicited, ontological commitments (i.e. ontologies) are used by various applications such as information integration and mediation of heterogeneous sources. Though ontologies can differ in structure and semantics, they all are build on a shared Lexon Base.

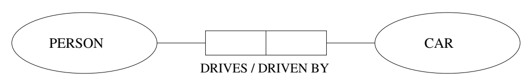

Figure 2: Illustration of a lexon that is described in a hypothetical context γ .

The commitments are specified in a designated language, called Ω-RIDL (Verheyden, De Bo, & Meersman 2004). It describes semantic constraints in terms of lexons, covering all classical database constraints (cfr. ORM). It also specifies which role/co-role label pairs are interpreted as which ontological relationship (such as subsumption, part-of). Consequently, this impacts the semantics of the commitment.

Commitments are also categorised and stored in a *commitment library* in the DOGMA server.

Contexts and Term Disambiguation

A lexon is a lexical representation of a conceptual relationship between two concepts, however, there is no bijective mapping between a lexical representation and a concept. Consider for example phenomenons such as synonyms and

[4]With respect to the application domain.

[5]NIAM (Verheijen & Van Bekkum 1982) is the predecessor of ORM (Halpin 2001).

[6]Although it is outside the scope of this paper, we find it valuable to note that in the research community it is debated that consistency is not necessarily a requirement for an ontology to be useful.

homonyms that can result in frustrating misunderstanding and ambiguity (see Def. 5). As we have seen, the meaning of a lexical term can vary depending on the context that holds.

In DOGMA, a context is used to group lexons that are related[7] to each other in the conceptualisation of a domain.

A context in DOGMA has one fundamental property: it is also a mapping function used to disambiguate terms by making them language-neutral. Based on Meersman (2001), we can give the following definition for a context:

Definition 2 *A context $\gamma \in \Gamma$ is a mapping $\gamma : T \cup R \rightarrow C$ from the set of terms and roles to the set of concept identifiers in the Universe of Discourse (UoD) C. In a context, every term or role is mapped to at most one concept identifier. A context γ is also a reference to one or more documents and/or parts of a document[8]. This reference is defined by the mapping $cd : \Gamma \rightarrow \mathcal{D}$.*

In this case we can check which lexons are valid in that specific context, more specifically those lexons extracted from (the parts of) the documents to which the context γ refers. A tuple $< \gamma, t >$ identifies a unique concept. With a concept we mean the thing itself to which we refer by means of a term (or role) in the Lexon Base. If we want to describe the set of concepts of our UoD formally, we can do this, according to Meersman (2001), by introducing the partial function $ct : \Gamma \times T \cup R \rightarrow C$ which associates a concept with a tuple consisting of a context and a term (or role). This partial function, which describes a form of *meaning articulation*, is defined as follows:

Definition 3 (meaning articulation) *Given the partial function $ct : \Gamma \times T \cup R \rightarrow C$, then*

$$ct(\gamma, t) = c \Leftrightarrow \gamma(t) = c.$$

An association $ct(\gamma, t) = c$ is called the "meaning articulation" or articulation[9] of a term t (in a particular context γ) into a concept identifier c. ct is called a meaning articulation mapping.

The set of concept identifiers C of the UoD can be formally defined as:

Definition 4 *The set of concepts identifiers $C = \{ct(\gamma, t) | \gamma \in \Gamma, t \in T \cup R\}$.*

Example 1 illustrates the two latter definitions:

Example 1 *Consider a term "capital". If this term was elicited from a typewriter manual, it has a different meaning than when elicited from a book on marketing. Hence, we have resp. two contexts: γ_1 = typewriter manual, and γ_2 = marketing book. To express that "capital" is associated with different meanings, we write $ct(\gamma_1, capital) = c_1$, and $ct(\gamma_2, capital) = c_2$.*

Until now, the endpoint of the meaning articulation is a meaningless concept identifier $c_1, c_2 \in C$. However, in the next section we will introduce the Concept Definition Server. Each concept identifier itself will point to a particular concept definition. The terms (on the *language level*) that are articulated (using ct) are then mapped to a particular *explication* of a meaning, i.e. a concept definition of a term residing in the Concept Definition Server (on the *conceptual level*), instead of to a meaningless concept identifier.

Before we continue, some useful terminology, as defined by De Bo and Spyns (2004), is presented in Def. 5:

Definition 5

- *Two terms $t_1 \in T$ and $t_2 \in T$ are synonyms within a context γ if and only if $(\gamma(t_1) = c \Leftrightarrow \gamma(t_2) = c)$.*
- *Two identical terms $t \in T$ are called homonyms if and only if $\exists \gamma_1, \gamma_2 \in \Gamma : \gamma_1(t) \neq \gamma_2(t)$.*

These definitions also hold for roles $r \in R$.

Completing the Articulation: Concept Definition Server

The idea for a Concept Definition Server (CDS) was first mentioned in (De Bo, Spyns, & Meersman 2004), and is based on the structure of Wordnet (Fellbaum 1998). CDS is a database in which you can query for a term, and get a set of different meanings or *concept definitions* (called *senses* in Wordnet) for that term. A concept definition is unambiguously explicated by a gloss (i.e. a natural language (NL) description) and a set of synonymous terms. Consequently we identify each concept definition in the CDS with a concept identifier $c \in C$.
The following definition specifies the CDS:

Definition 6 *We define a Concept Definition Server Υ as a triple $< T_\Upsilon, \mathcal{D}_\Upsilon, concept >$ where:*

- *T_Υ is a non-empty finite set of strings (terms)[10];*
- *$\mathcal{D}_\Upsilon$ is a non-empty finite document corpus;*
- *$concept : C \longmapsto \mathcal{D}_\Upsilon \times \wp(T_\Upsilon)$ is an injective mapping between concept identifiers $c \in C$ and concept definitions.*

Further, we define $conceptdef(t)$

$$= \{concept(c) \,|\, concept(c) =< g, sy > \wedge t \in sy\},$$

where gloss $g \in \mathcal{D}_\Upsilon$ and synset $sy \subseteq T_\Upsilon$.

Going from the language level to the conceptual level corresponds to articulating lexons into meta-lexons:

Definition 7 *Given a lexon $l :=< \gamma, t_1, r_1, r_2, t_2 >$, and an instance of an articulation mapping $ct : \Gamma \times T \cup R \rightarrow C$ with $ct(\gamma, t_1) = c_{t_1}, ct(\gamma, r_1) = c_{r_1}, ct(\gamma, r_2) = c_{r_2}, ct(\gamma, t_2) = c_{t_2}$ ($c_{t_1}, c_{r_1}, c_{r_2}, c_{t_2} \in C$). A meta-lexon $m_{l,ct} :=< c_{t_1}, c_{r_1}, c_{r_2}, c_{t_2} >$ (on the conceptual level) is the result of "articulating" lexon l via ct.*

[7] Not necessarily in a logical way but more in an informal way. E.g., lexons are related because they were elicited from the same source, i.e. the elicitation context.

[8] At this stage, we only require a document should provide information what or whom the lexon was elicited from. See our case study below for a concrete example.

[9] We adopt the term articulation from Mitra et al. (2000) (see discussion).

[10] Additionally, we could require $T \cup R \subseteq T_\Upsilon$ (T and R from the Lexon Base). Doing so, we require each term and role in the Lexon Base to be a term in the synset of at least one concept definition.

In Fig. 3 the articulation is illustrated by a *meaning ladder* going from the (lower) language level to the (higher) conceptual level and vice-versa. This ladder is inspired by Stamper's *semiotic ladder*. Stamper (1973) argues that it is naive to see information as a primitive or atomic concept. From his operational point of view he means that in defining something, it is important to specify precisely by what procedure or operations to measure or perform. Hence, the solution in attempting to define "information" is to see information as signs and to define the different aspects or levels of these signs based on the operations one can do on these signs. His semiotic ladder consists of six views on signs (levels) from the perspective of physics, empirics, syntactics, semantics, pragmatics and the social world, that together form a complex conceptual structure. We refer to Fig. 1, where we introduced the levels and the ladder in the application–elicitation setting.

Next, another interesting definition can be given:

Definition 8 (Meta-lexon Base) *Given a Lexon Base* Ω *and a total articulation mapping* $ct : \Gamma \times T \cup R \rightarrow C$*, a Meta-lexon Base* $M_{\Omega,ct} = \{m_{l,ct} | l \in \Omega\}$ *can be induced.*

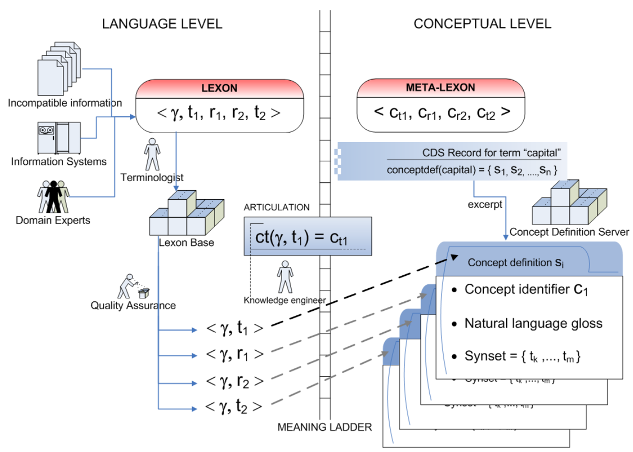

Figure 3: Illustration of the two levels in DOGMA ontology: on the left – the lexical level, lexons are elicited from various contexts. On the right, there is the conceptual level consisting of a concept definition server. The meaning ladder in between illustrates the articulation of lexical terms into concept definitions.

Example 2 *As an illustration of the defined concepts, consider Fig. 4. The term "capital" in two different contexts can be articulated to different concept definitions in the CDS. The terms are part of some lexons residing in the Lexon Base. The knowledge engineer first queries the CDS* Υ *for the various concept definitions of the term:* $conceptdef(capital) = S_{capital} \subseteq \mathcal{D}_{\Upsilon} \times \wp(T_{\Upsilon})$*. Next, he articulates each term to the concept identifier of the appropriate concept definition:*

- *Term "capital" was extracted from a typewriter manual, and is articulated to concept identifier* c_1 *that corresponds to concept definition (or meaning)* $s_1 \in S_{capital}$ *(as illustrated on the right of Fig. 4). A gloss and set of synonyms (synset) is specified for* s_1*:*

$$concept\big(ct(typewriter\,manual, capital)\big) = s_1.$$

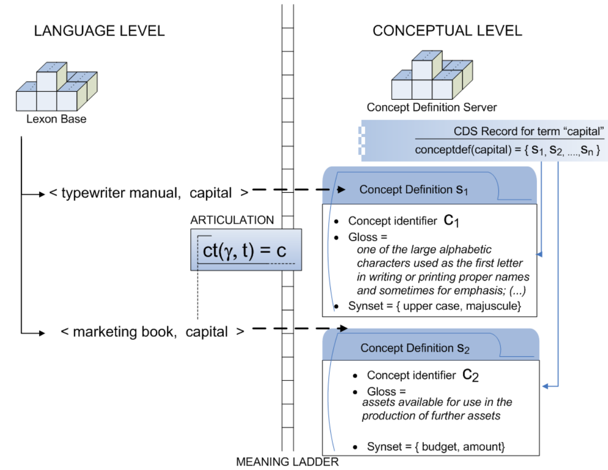

Figure 4: Illustration of two terms (within their resp. contexts), being articulated (via the mapping ct) to their appropriate concept definition.

- *Term "capital" was extracted from a marketing book, due to the different context it was extracted from, it is articulated to another concept identifier* c_2 *that is associated with a concept definition* $s_2 \in S$*:*

$$concept\big(ct(marketing\,book, capital)\big) = s_2.$$

On the other hand, suppose we have elicited a term "exercise" from the typewriter manual, and a term "example" from the marketing book. The engineers decide independently to articulate the resp. terms to the same concept definition with concept identifier c_3 *with gloss: "a task performed or problem solved in order to develop skill or understanding":*

$$\begin{aligned} c_3 &= ct(typewriter\,manual, exercise) \\ &= ct(marketing\,book, example). \end{aligned}$$

This articulation defines a semantic bridge between two terms in two different ontological contexts.

Shared Competency Ontology-Building

In this section we illustrate context-driven ontology elicitation in a realistic case study of the European CODRIVE[11] project.

Competencies and Employment

Competencies describe the skills and knowledge individuals should have in order to be fit for particular jobs. Especially in the domain of vocational education, having a central shared and commonly used competency model is becoming crucial in order to achieve the necessary level of

[11] CODRIVE is an EU Leonardo da Vinci Project (BE/04/B/F/PP-144.339).

interoperability and exchange of information, and in order to integrate and align the existing information systems of competency stakeholders like schools or public employment agencies. None of these organisations however, have successfully implemented a company-wide “competency initiative”, let alone a strategy for inter-organisational exchange of competency related information.

The CODRIVE project aims at contributing to a competency-driven vocational education by using state-of-the-art ontology methodology and infrastructure in order to develop a conceptual, shared and formal KR of competence domains. Domain partners include educational institutes and public employment organisations from various European countries. The resulting shared “Vocational Competency Ontology” will be used by all partners to build interoperable competency models.

In building the shared ontology, the individual ontologies of the various partners need to be aligned *insofar necessary*. It is important to realise that costly alignment efforts only should be made when necessary for the shared collaboration purpose. In order to effectively and efficiently define shared relevant ontological meanings, context is indispensable.

Example: Adding a Term

The example concerns two participating organisations EI and PE, instances of resp. *educational institute* and *public employment agency*. Core shared and individual ontologies have already been defined for both EI and PE. Fig. 5 illustrates the different contexts[12] called resp. $SHARED$, EI and PE. The $SHARED$ ontology has amongst its concepts $Task$, with as subtypes $Educational\,Task$ and $Job\,Task$ (is_a represented by lexons). The concepts as referred in lexons are in fact terms, but within the context $SHARED$ they refer to at most one concept definition. The concepts underlined in the rules below are also modelled but not shown, similarly for the specialisation hierarchies of EI and PE.

EI is a national bakery institute, responsible for defining curriculum standards for bakery courses. It now wants to add a new term “panning” to the ontology. It defines this informally as the act of “depositing moulded dough pieces into baking pans with their seam facing down”. Fig. 5 shows the steps in ontology alignment: step 1 is adding the new term (which resides in some lexon which is not shown) to ontology EI. Step 2 is the meaning articulation, illustrated by an arrow going from the language level of the ontology to a concept definition in the CDS.

Step 3 is triggered by the following informal rule:

R1: The CODRIVE ontology server (COS) asks EI to classify the shared concept to which the term belongs.

The EI representative classifies panning as an $Educational\,Task$ (illustrated as an arrow labelled with step 3). COS now looks in its ontological meta-model. One of the rules there demands:

R2: IF a New Task is an Educational Task, and the Individual Ontology Owner is an Educational Institute THEN a Full semantic analysis of the New Task needs to be added to the Individual Ontology of the Individual Ontology Owner;

another meta-rule fires as an immediate consequence:

R3: IF a Full Semantic Analysis needs to be made of a Concept in an Individual Ontology or Shared Ontology THEN the Concept Template needs to be filled out in that Ontology. Furthermore, for each Term and Role of that definition, a Meaning Articulation needs to be defined.

This means that in this case the template states it is necessary to know who is the performer of the task (e.g. $Student$), what inputs are necessary for the task (e.g. Pan, $Dough$), what is the result of the task ($Pan\,with\,Dough$), and so on. Rules **R2** and **R3** trigger step 4: in the EI context, the new task $Panning$ is semantically analysed, which boils down to extending the description in terms of lexons (illustrated by the lexons within the dashed box). Similarly to step 1, each new term or role in this box must be articulated (not shown in the figure).

Concurrently another rule triggers step 5:

R4: IF an Educational Task is added to an Individual Ontology THEN a corresponding Job Task needs to be defined in all instances of Individual Ontology of all Public Employment Agencies;

The rationale for this rule is that public employment agencies need to be aware of changes to the curricula of educational institutes, so that they are better able to match job seekers with industry demands. However, unlike the definitions of educational tasks, the job task definitions in

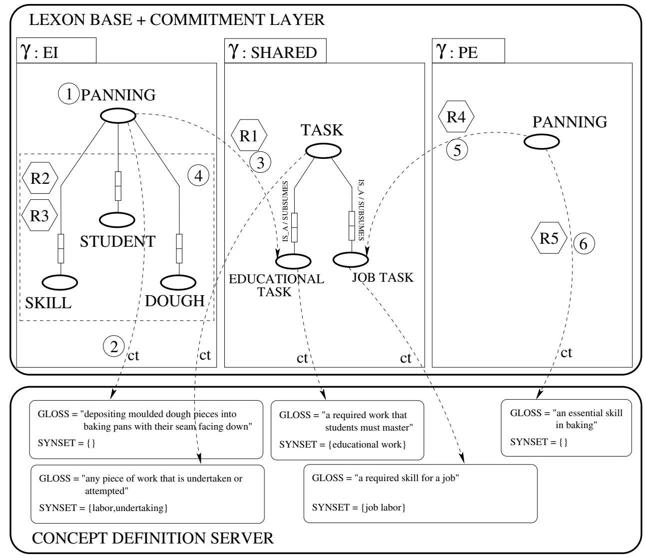

Figure 5: Illustration of the case study: top level, from right to left: ontologies EI, $SHARED$, and PE. On the bottom: CDS.

[12]In this case study, each context corresponds to exactly one ontology and vice-versa. However, an ontology engineer might select lexons from various contexts for modelling his ontology.

public employment agency ontologies only require a short informal description of the concept itself, not an extended template definition (step 6):

`R5: IF a JobTask is added to an IndividualOntology THEN a Gloss needs to be defined for that Concept.`

Of course, public employment agencies also could have the need for template definitions, but those would refer to the job matching processes in which the tasks play a role (*why* is panning needed), not to *how* the tasks themselves are to be performed.

Note the density of lexon elicitation in the EI ontology (on the left of Fig. 5) compared to the sparsely populated PE ontology (on the right of Fig. 5). The density reflects the minimal level of modelling details needed. Our context-driven ontology elicitation avoids wasting valuable modelling time and enormous cost.

This real world example illustrates one simple problem, viz. identical terms in different contexts can have different meanings (homonyms). However, disambiguation can become very complex when considering e.g., synonymy.

Discussion and Future Directions

Shamsfard et al. (2003) provide a comprehensive survey of methods and tools for (semi-)automatic ontology elicitation. However, in this paper our focus is not on automation. Work which is strongly related with what we need is e.g., Mitra et al. (2000), who indirectly adopt some of those features we concluded with in our synthesis earlier. They illustrate a semi-automatic tool for creating a mapping between two ontologies (in fact contexts). Their motivation is that two different terms can have the same meaning and the same term can have different meanings, which exactly defines the lexical disambiguation problem. This mapping is manifested by an *articulation ontology*, which is automatically generated from a set of *articulation rules* (i.e. semantic relationships) between concepts in each context resp.

In our framework, the CDS provides a basic means for relevant alignment of heterogeneous ontologies. The concept definitions (gloss, synset) in the CDS support the meaning articulation of language level terms. As was illustrated in Ex. 2, the articulation of terms from different contexts to a shared CDS, results in cross-context equivalence relations, i.e. synonyms.

The meta-rules we made in step 3 and 5, were not formalised explicitly in this paper. However, we could define a syntax, e.g.:

$$\begin{aligned} ct(EI, panning) &\preceq ct(SHARED, educational\, task) \\ ct(PE, panning) &\preceq ct(SHARED, job\, task) \end{aligned}$$

The semantics of the relation $\preceq$ is comparable to McCarthy's lifting rules: it allows us to specify an alignment between a term in one context to a possibly more general term in another context. In the future we will extend this feature and provide a formal semantics of meta-rules. This can be very powerful in context-driven ontology elicitation and application such as meta-reasoning on context and ontology alignment processes, and meaning negotiation processes between stakeholders. Currently we are exploring reasoning for commitment analysis and conceptual graph tools for ontology elicitation and analysis.

Initially, the lexon base and CDS are empty, but the CDS can be easily populated by importing similar information from publically available electronic lexical databases, such as Wordnet (Fellbaum 1998) or Cyc (OpenCyc). The Lexon Base is populated during the first step of an ontology elicitation process by various (not necessarily human) agents. See Reinberger & Spyns (2005) for unsupervised text mining of lexons. The second step in the elicitation process is to articulate the terms in the "learned" lexons.

Finally, we note that in the case study we did not consider the semantic constraints completely. The only visible constraint is the interpretation of the role/co-role pair is_a/subsumes as the ontological specialisation relation in ontology $SHARED$. Adding other relations such as defined in conceptual graph theory (Sowa 2000), will considerably improve the power of our meaning articulation approach.

Conclusion

We have presented an extension to the DOGMA ontology framework that enables context-driven ontology elicitation. We introduced contexts and a concept definition server to (i) logically group knowledge, (ii) to disambiguate the lexical representation of concepts and relationships, by distinguishing between language level and conceptual level, and (iii) to build semantic bridges between different ontological contexts. Next, we illustrated context-driven ontology elicitation considering a real world case example. Finally, we summarised related work and showed how our work can be extended.

References

Barwise, J., and Perry, J. 1983. *Situations and Attitudes*. MIT Press.

Buchler, J. 1955. *Philosophical Writings of Peirce*. New York: Dover Publ.

Buvač, S., and Fikes, R. 1995. A declarative formalization of knowledge translation. In *Proc. of 4th Int'l Conf. on Information and Knowledge Management (ACM CIKM 95)*.

Buvač, S. 1996a. Quantificational logic of context. *AAAI/IAAI* 1:600–606.

Buvač, S. 1996b. Resolving lexical ambiguity using a formal theory of context. In Van Deemter, K., and Peters, S., eds., *Semantic Ambiguity and Underspecification*. CSLI Publications.

De Bo, J.; Spyns, P.; and Meersman, R. 2004. Assisting ontology integration with existing thesauri. In *Proc. of On the Move to Meaningful Internet Systems (OTM2004) (Ayia Napa, Cyprus)*, 801–818. Springer Verlag.

De Leenheer, P., and Meersman, R. 2005. Towards a formal foundation of DOGMA ontology: part I. Technical Report STAR-2005-06, VUB STARLab.

Farquhar, A.; Dappert, A.; Fikes, R.; and Pratt, W. 1995. Integrating information sources using context logic. In Knoblock, C., and Levy, A., eds., *Information Gathering from Heterogeneous, Distributed Environments*.

Farquhar, A.; Fikes, R.; and Rice, J. 1997. The ontolingua server: a tool for collaborative ontology construction. *Int'l Journal of Human-computer Studies* 46(6):707–727.

Fellbaum, C., ed. 1998. *Wordnet, an Electronic Lexical Database*. MIT Press.

Giunchiglia, F. 1993. Contextual reasoning. *special issue on I Linguaggi e le Macchine* XVI:345–364.

Gruber, T. 1993. Cyc: a translation approach to portable ontologies. *Knowledge Acquisition* 5(2):199–220.

Guarino, N. 1998. Formal ontology and information systems. In *Proc. of the 1st Int'l Conf. on Formal Ontologies in Information Systems (FOIS98) (Trento, Italy)*, 3–15. IOS Press.

Guha, R., and D., L. 1990. Cyc: a midterm report. *AI Magazine* 11(3):32–59.

Guha, R. V. 1991. Contexts: a formalization and some applications. Technical Report STAN-CS-91-1399, Stanford Computer Science Department, Stanford, California.

Halpern, J., and Moses, Y. 1992. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence* 54:319–379.

Halpin, T. 2001. *Information Modeling and Relational Databases (From Conceptual Analysis to Logical Design)*. Morgan Kauffman.

Hintikka, J. 1963. The modes of modality. In *Acta Philosophica Fennica, Modal and Many-valued Logics*, 65–81.

Kripke, S. 1963. Semantic analysis of modal logic i. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9:67–96.

Langacker, R. 1987. *Foundations of Conditional Grammars*. Stanford University Press.

Lenat, D. 1995. Cyc: a large-scale investment in knowledge infrastructure. *Communications of the Association for Computing Machinery* 38(11).

McCarthy, J., and Buvač, S. 1994. Formalizing context (expanded notes). Technical Report STAN-CS-TN-94-13, Stanford University.

McCarthy, J. 1987. Generality in artificial intelligence. *Communication of the Association for Computing Machinery* 30(12):1030–1035.

McCarthy, J. 1993. Notes on formalizing context. In *Proc. of the 15th Int-l Joint Conf. Artificial Intelligence (IJCAI93) (Chambry, France)*, 555–560. Morgan Kaufmann.

McCarthy, J. 1996. A logical ai approach to context. Unpublished note.

Meersman, R. 1999. The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems. In *Proc.of the Conf. on Cooperative Database Systems (CODAS99)*, 1–14. Springer Verlag.

Meersman, R. 2001. Reusing certain database design principles, methods and techniques for ontology theory, construction and methodology. Technical report, VUB STAR Lab, Brussel.

Mineau, G., and Gerbé, O. 1997. Contexts: A formal definition of worlds of assertions. In *Proc.of the 5th Int'l Conf. on Conceptual Structures (ICCS-97)(Seattle, USA)*, 80–94. Springer Verlag.

Mineau, G.; Missaoui, R.; and Godinx, R. 2000. Conceptual modeling for data and knowledge management. *Data and Knowledge Engineering* 33(2):137–168.

Mitra, P.; Wiederhold, G.; and Kersten, M. L. 2000. A graph-oriented model for articulation of ontology interdependencies. In *EDBT '00: Proceedings of the 7th International Conference on Extending Database Technology*, 86–100. London, UK: Springer-Verlag.

Nayak, P. 1994. Representing multiple theories. In *Proc. of the 12th Nat'l Conf. on Artificial Intelligence (AAAI 94)(Seattle, Washington)*. AAAI Press.

OpenCyc. http://www.opencyc.org.

Reinberger, M.-L., and Spyns, P. 2005. Unsupervised text mining for the learning of DOGMA-inspired ontologies. In *Buitelaar P., Handschuh S., and Magnini B.,(eds.), Ontology Learning and Population*, in press. IOS Press.

Shamsfard, M., and Barforoush, A. A. 2003. The state of the art in ontology learning: a framework for comparison. *the Knowledge Engineering Review* 18(4):293–316.

Sowa, J. 1995. Syntax, semantics, and pragmatics of contexts. In *Proc. of the 3rd Int'l Conf. on Conceptual Structures: Applications, Implementation and Theory*, 1–15. Springer-Verlag.

Sowa, J. 1997. Peircean foundations for a theory of context. In *Conceptual Structures: Fulfilling Peirce's Dream*, 41–64. Springer-Verlag.

Sowa, J. 2000. *Knowledge Representation - Logical, Philosophical and Computational Foundations*. Brooks/Cole Publishing Co.

Stamper, R. 1973. *Information in Business and Administrative Systems*. John Wiley and Sons.

Theodorakis, M. 1999. *Contextualization: an Abstraction Mechanism for Information Modeling*. Ph.D. Dissertation, University of Crete, Greece.

Ushold, M., and Gruninger, M. 1996. Ontologies: Principles, methods and applications. *The Knowledge Engineering Review* 11(2):93–136.

Verheijen, G., and Van Bekkum, J. 1982. Niam, an information analysis method. In *Proc. of the IFIP TC-8 Conference on Comparative Review of Information System Methodologies (CRIS 82)*. North-Holland.

Verheyden, P.; De Bo, J.; and Meersman, R. 2004. Semantically unlocking database content through ontology-based mediation. In *Proc. of the 2nd Workshop on Semantic Web and Databases, VLDB Workshops (SWDB 2004) (Toronto, Canada)*, 109–126. Springer-Verlag.