HEAVYWEIGHT ONTOLOGY MATCHING
A Method and a Tool Based on the Conceptual Graphs Model

Frédéric Furst
LARIA, University of Amiens, UPJV, 33 rue Saint Leu - 80039, Amiens, France

Francky Trichet
LINA, University of Nantes, 2 rue de la Houssinière BP 92208 - 44322, Nantes, France

Keywords: Heavyweight Ontology, Axioms, Graph-Based Techniques, Ontology Matching, Conceptual Graphs.

Abstract: Managing multiple ontologies is now a core question in most of the applications that require semantic interoperability. The Semantic Web is surely the most significant application of this report: the current challenge is not to design, develop and deploy domain ontologies but to define semantic correspondences among multiple ontologies covering overlapping domains. In this paper, we introduce a new approach of ontology matching named *axiom-based ontology matching*. As this approach is founded on the use of axioms, it is mainly dedicated to heavyweight ontologies (an *heavyweight ontology* is a *lightweight ontology*, *i.e.* an ontology simply based on a hierarchy of concepts and a hierarchy of relations, enriched with axioms used to fix the semantic interpretation of concepts and relations), but it can also be applied to lightweight ontologies as a complementary approach to the current techniques based on the analysis of natural language expressions, instances and/or taxonomical structures of ontologies. This new matching paradigm is defined in the context of the Conceptual Graphs model (CG), where the projection (*i.e.* the main operator for reasoning with CG which corresponds to homomorphism of graphs) is used as a means to semantically match the concepts and the relations of two ontologies through the explicit representation of the axioms in terms of conceptual graphs. We also introduce an ontology of representation dedicated to the reasoning of heavyweight ontologies at the meta-level.

1 INTRODUCTION

Ontology matching is at the heart of the multiple-ontology management process that is now a core question in most of the applications that require semantic interoperability such as the Semantic Web (Doan and Halevy, 2005; Noy, 2004; Shvaiko and Euzenat, 2005).

The strategies for matching ontologies are quite diverse: hierarchical clustering techniques, Formal Concept Analysis, analysis of terminological features of concepts and relations (*i.e.* names or natural-language definitions) or analysis of structure. However, as recalled in (Gomez-Perez et al., 2003), most of the works that deal with ontology alignment only consider *lightweight ontologies*, *i.e.* ontologies simply composed of taxonomies of concepts and taxonomies of relations. The most significant example of this situation is the benchmark used during the campaigns "Ontology Alignment Evaluation Initiative" (Ashpole et al., 2005; Benjamins et al., 2006) (http://oaei.ontologymatching.org/), where the ontologies used for the experiments are only lightweight ones: for instance, the anatomy real world case used in 2006 covers the domain of body anatomy and consists of two ontologies with an approximate size of several 10k classes and several dozens of relations, but none of these two ontologies includes axioms. Axioms are the main building blocks for fixing the semantic interpretation of the concepts and the relations, and this is what differentiates *lightweight ontologies* from *heavyweight ontologies*. Of course, currently, there are not so many real-world ontologies that make substantial use of axioms. However, as introduced by T. Berners-Lee (Berners-Lee et al., 2001) - "*For the semantic web to function, computers must have access to structured collections of information and sets of inference rules that they can use to conduct automated reasoning*" - we think that the need to develop *heavyweight ontologies*, *i.e.* ontologies

Furst F. and Trichet F. (2007).
HEAVYWEIGHT ONTOLOGY MATCHING - A Method and a Tool Based on the Conceptual Graphs Model.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - AIDSS*, pages 265-270
DOI: 10.5220/0002348802650270

which include axioms used both to represent all the semantics of a domain D and to conduct automated reasonings on assertions of D (more precisely, to ensure that the correct interpretation to semantics of a construct will be given at run time, or in logic jargon to restrict possible interpretations of the construct in a domain of discourse), will inevitably increase in an immediate future; this is also clearly demonstrated by the current W3C trend which aims at standardising a Semantic Web Rule Language.

The work presented in this paper aims at defining a new ontology matching approach based on the explicit use of all the components of a heavyweight ontology. This approach requires the explicit representation of the axioms of the two ontologies (that are considering for the matching process) at the conceptual level, and not at the operational level as it is usually the case in most of the works related to ontological engineering: for instance in Protégé (Noy, 2004), the axioms are directly represented in an operational form (*i.e.* rules or constraints with fixed and predefined operational semantics) by using the PAL language based on logical expressions.

To represent heavyweight ontologies at the conceptual level, we use OCGL (*Ontology Conceptual Graphs Language*) (Fürst et al., 2004). This modelling language is based on a graphical syntax inspired from those of the Conceptual Graphs model (CGs). The CGs model, first introduced by Sowa (Sowa, 1984), is an operational knowledge representation model which belongs to the field of semantic networks. This model is mathematically founded both on logics and graph theory. Two approaches for reasoning with CGs can be distinguished: (1) considered CGs as a graphical interface for logics and reasoning with logic and (2) considered CGs as a graph-based knowledge representation and reasoning formalism with its own reasoning capabilities. In our work, we adopt the second approach by using the projection (a graph-theoretic operation corresponding to homomorphism) as the main reasoning operator; projection is sound and complete w.r.t. deduction in FOL. The CG model allows us to represent terminological knowledge through the specification of concepts and relations, and to represent both classical properties (such as subsumption or algebraic properties) and any kind of axioms at the conceptual level. This explicit graph-based representation of axioms coupled with reasoning capabilities based on graphs homomorphism facilitates the topological comparison of axioms. The matching method we propose mainly relies on this feature: ontology morphism founded on graph-based knowledge representation and graph-based reasoning mechanisms.

The rest of this paper is organized as follows. Section 2 presents the modelling paradigm we advocate for defining a domain ontology. Section 3 introduces the basic foundations of our axiom-based matching method and presents the principles of our algorithm. Section 4 compares our approach to related work.

2 CONTEXT OF THE WORK

The OCGL modelling language (*Ontology Conceptual Graphs Language*) we use for specifying an ontology (at the conceptual level) is based on three building blocks: Concepts, Relations and Axioms. Representing an ontology in OCGL mainly consists in (1) specifying the conceptual vocabulary of the domain and (2) specifying the semantics of this conceptual vocabulary through axioms (Fürst et al., 2004). The conceptual vocabulary consists of a set of **Concepts** and a set of **Relations**. These sets can be structured by using both well-known conceptual properties called Schemata Axioms and Domain Axioms. The union of these Schemata Axioms and Domain Axioms corresponds to what we call **Axioms**.

The **Schemata Axioms** proposed in OCGL are: (1) the *ISA* link between two concepts or two relations (subsumption property) used to construct concept/relation taxonomies (tree or lattice), (2) the *Abstraction* of a concept, which corresponds to an *Exhaustive-Decomposition* in some works (Gomez-Perez et al., 2003), (3) the *Disjointness* of two concepts, (3) the *Signature* of a relation, (4) the *Algebraic properties* of a relation (symmetry, reflexivity, transitivity, irreflexivity, etc.), (5) the *Exclusivity* or the *Incompatibility* between two relations (the incompatibility between R_1 and R_2 is formalized by $\neg(R_1 \wedge R_2)$, the exclusivity is formalized by $\neg R_1 \Rightarrow R_2$) and finally (6) the *Cardinalities* of a relation.

Domain Axioms correspond to knowledge which can not be represented with Schemata Axioms (representing classical properties of concepts or relations). The OCGL graphical syntax used to express such an axiom is based on the Conceptual Graphs model. Thus, an axiom is composed of an *Antecedent part* and a *Consequent part*, with a formal semantics that intuitively corresponds to: *if the Antecedent part is true, then the Consequent part is true*. Figure 1 shows the OCGL graph representing the axiom "*The enemy of my friend is my enemy*". Note that this axiom is a real Domain Axiom because it cannot be represented by using classical properties, in comparison with the axiom "*The friend of my friend is my friend*" which is represented by the transitivity of the relation called `Friend(Human,Human)`, that is a Schemata Axiom.

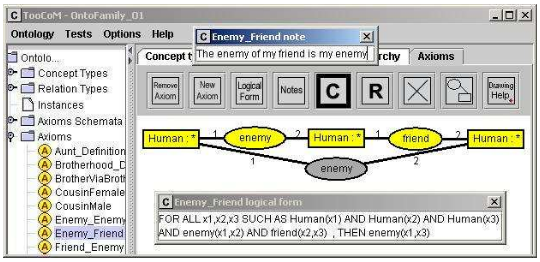

Figure 1: Representation of an axiom in TooCom. The bright nodes represent the antecedent part, the dark ones the consequent part. A concept node (indicated by a rectangle) is described by a label and a marker that identifies the considered instance (the marker $*$ denotes an undefined instance). A relation node (indicated by an ellipse) is only described by a label. An edge between a concept and a relation is labeled with the position of the concept in the signature of the relation. The logical expression of the graph is automatically generated.

OCGL has been implemented in a tool, called TooCom (*a Tool to Operationalize an Ontology with the Conceptual Graph Model*), dedicated to the edition and operationalization of domain ontologies (Fürst and Trichet, 2005b; Fürst and Trichet, 2005a). TooCom is available under GNU GPL license at the following URL: `http://sourceforge.net/projects/toocom/`.

3 AXIOM-BASED SEMANTIC MATCHING

The objective of ontology matching is to discover and evaluate semantic links (*e.g.* identity or subsumption) between conceptual primitives (concepts and relations) of two given ontologies supposed to be built on related domains. Our approach relies on the use of the axiomatic level of the ontologies to discover semantic analogies between primitives, in order to reveal identities between them and to calculate the similarity coefficient of these identities, *i.e.* a coefficient that indicates how closely two concepts or relations are related. Of course, using the axiomatic level does not forbid to use the terminological level; these two approaches complement each other. Our algorithm (implemented in the current version of TooCom) takes as input two ontologies O_1 and O_2 (represented in OCGL) and provides as output potential similarity between two concepts or two relations: the result is a set of matchings (P_i, P'_j, C), where P_i and P'_j are respectively conceptual primitives (concepts and relations) of O_1 and O_2, and C the similarity coefficient between P_i and P'_j. Of course, for a given primitive P_i of O_1, several (or any) matchings can exist with primitives of O_2, and vice versa. Both *Schemata Axioms* and *Domain Axioms* are used to evaluate or discover primitive matchings.

First, in order to allow the end-user to refine the results of our algorithm according to the matching context, we have associated a weight to each OCGL property. These weights can be modified in order to modulate their influence on the evaluation of the matching. Thus, there are parameters of our algorithm which can be changed to improve the precision of the results. By default, the values of the weights are ordered as follows: $W_{AlgebraicProperties}(W_{Sym}, W_{Trans}, W_{Refl}, W_{Irref}, W_{AntiSym}) > W_{Disjointness} = W_{Incompatibility} = W_{Exclusivity} > W_{Cardinality_{min}} = W_{Cardinality_{max}} > W_{Axiom} > W_{Signature} > W_{Abstraction} > W_{ISA}$. Again, this scale of weights is a just a guess which for us corresponds to a universal distribution for all ontologies; it can be modified by the end-user according to the kind of ontologies which are considered and/or subjective preferences.

Then, to detect analogies between axioms represented as graphs, and then to detect analogies between the primitives corresponding to the nodes of the graphs, the Domain Axioms are transcribed into a more abstract form, that preserves the topological structures of the graphs. These abstract representations are based on an ontology of representation called **MetaOCGL**.

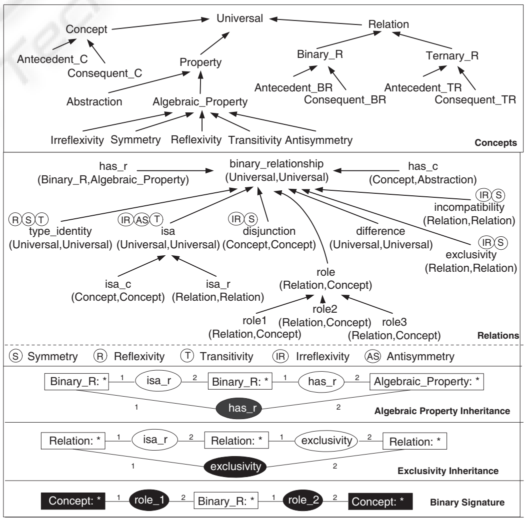

Figure 2: Concepts, relations, Schemata Axioms and Domain Axioms of MetaOCGL.

MetaOCGL is the ontology of the OCGL language, expressed in OCGL. MetaOCGL can then be considered as an ontology at the meta-level (Gomez-Perez et al., 2003). As shown in figure 2, MetaOCGL includes (1) Concepts, (2) Relations, (3) Schemata Axioms and (4) Domain Axioms.

The *MetaOCGL Concepts* are used to represent the OCGL primitives: *Concept* with its two sub-primitives *Antecedent-C* and *Consequent-C* used in the context of an OCGL Domain Axiom, *Property* which includes the *Algebraic-Properties* of a OCGL relation and the *Abstraction* of a OCGL concept and *Relation* which again includes the Antecedent/Consequent point of view for the different kinds of OCGL relations (*Binary-R*, *Ternary-R*, etc.). The *MetaOCGL Relations* are used to represent the links between the OCGL primitives: *isa* relation which can be stated between two OCGL concepts or two OCGL relations - the signature is *(Universal,Universal)*, *exclusivity/incompatibility* between OCGL relations, *disjointness* of OCGL concepts, links between OCGL relations and concepts in a graph that expresses an OCGL Domain Axiom (*type-identity*, *difference*, *role*). The *MetaOCGL Schemata Axioms* are mainly used for describing the properties of the OCGL relations such as, for instance, the algebraic properties of the *isa* relationship (*Irreflexivity*, *Antisymmetry* and *Transitivity*). Finally, the *MetaOCGL Domain Axioms* are used to express the formal semantics of OCGL (for instance, the *Algebraic Property Inheritance* or the *Signature Conformity* presented in figure 2).

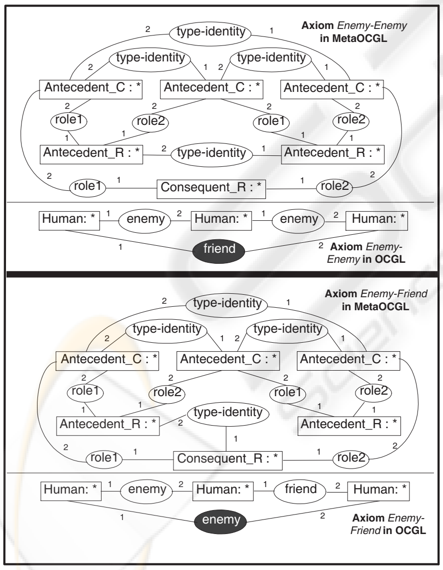

Figure 3: Two axioms of OntoFamily represented with MetaOCGL. The *type_identity*. links denote the fact that the nodes of the Domain Axiom (at the domain level) are similar, *i.e.* they have the same type. The two graphs (at the meta-level) are similar without considering type-identity links, but they differ when considering these links, because the relations of the antecedent part of the Domain Axiom ”Enemy Enemy” (at the domain level) have the same type, but not those of the Domain Axiom ”Enemy Friend”.

A domain ontology can be represented as a MetaOCGL instance (*i.e.* a MetaOCGL graph), as domain facts can be represented by OCGL graphs. The MetaOCGL graph that represents an ontology contains a part which is dedicated to the representation of the concept hierarchy, a part which is dedicated to the representation of the relation hierarchy, and as many parts as axioms in the ontology. Figure 3 shows the MetaOCGL graphs dedicated to the representation of the two axioms of OntoFamily O_1 *“The enemy of my enemy is my friend”* and *“The enemy of my friend is my enemy”*, and their corresponding meta-graphs in MetaOCGL. The MetaOCGL representation of an ontology expressed in OCGL is automatically provided by TooCom.

The comparisons between axioms represented in MetaOCGL are performed by using the *projection* operator of the Conceptual Graphs model, a graph-theoretic operation corresponding to homomorphism which is sound and complete w.r.t. deduction in FOL. A projection from a graph $G1$ into a graph $G2$ is a specific morphism of graphs which may restrict the labels of the vertices; it corresponds to a logical implication between $G1$ and $G2$. The figure 4 presents an example of projection.

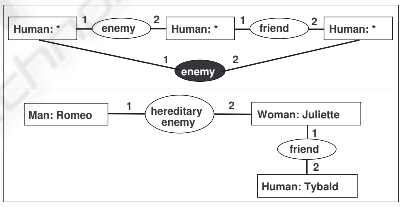

Figure 4: An example of projection between the Antecedent part of an axiom and a graph. The axiom is: ”The enemy of my friend is my enemy”. Its Antecedent part (white nodes of G1 presented at the top of the figure) can be projected into the graph G2 (the bottom of the figure), because each node of G1 has a corresponding node (in G2) that is more specific than itself: (Human:*) of G1 is more general than (Man:Romeo) of G2; (enemy) of G1 is more general than (hereditary enemy) of (G2); etc. In this context, there exists a projection from G1 into G2. Thus, the axiom can be applied to G2 to produce the following conclusion: ”Romeo is the enemy of Tybald”.

Given two graphs G_1 and G_2, which represent in MetaOCGL two axioms A_1 and A_2, if two projections exist from G_1 into G_2 and from G_2 into G_1, then A_1 and A_2 have the same structure. In this case, the axioms A_1 and A_2 express the same type of property, and the analogy between the two axioms can be extended to the primitives that appear in the axioms.

3.1 Algorithm: Principles

3.1.1 Using Schemata Axioms

Schemata Axioms that deal with only one primitive (*i.e.* algebraic properties and abstractions) are compared from O_1 to O_2, in order to discover primitive matchings. If an algebraic property (resp. an abstraction) appears in O_1 for a primitive p_1 and in O_2 for a primitive p_2, the coefficient c of the matching (p_1, p_2, c), if it exists, is increased by W_{Alg} (resp. W_{Abs}). If the matching does not exist, (p_1, p_2, W_{Alg}) (resp. (p_1, p_2, W_{Abs})) is created. If an algebraic property (resp. an abstraction) appears in O_1 for p_1 but not in O_2 for p_2 (or inversely), the coefficient c of the matching (p_1, p_2, c), if it exists, is decreased by W_{Alg} (resp. W_{Abs}). If it does not exist, $(p_1, p_2, -1 * W_{Alg})$ (resp. $(p_1, p_2, -1 * W_{Abs})$) is created. A partition (a *partition* (Gomez-Perez et al., 2003) is the combination of the abstraction of a concept (the head) and the disjointness of its children) is a property which is more semantically rich than a simple abstraction. So, if two concepts c_1 and c_2 are respectively the head concept of a partition in O_1 and O_2, the coefficient c of the matching (c_1, c_2, c), if it exists, is increased by $2 * W_{Abs}$ (or decreased by $2 * W_{Abs}$ if only one concept is involved in a partition). If it does not exist, $(c_1, c_2, 2 * W_{Abs})$ (or $(c_1, c_2, -2 * W_{Abs})$) is created.

Schemata Axioms that deal with two primitives (*i.e.* disjointness, incompatibility and exclusivity) are used either to modify the coefficients of existing matchings, or to create new ones. The coefficient of a matching whose two primitives are involved in a disjointness, an incompatibility or an exclusivity is increased by the corresponding weight (*i.e.* W_{Disj}, W_{Incomp} or W_{Exclu}). It is decreased if only one of the primitive is part of such a property. The matching is created with the corresponding coefficient if it does not exist.

Finally, table 1 presents the different actions that are done when considering the cardinalities. If the matching between the two considered relations does not exist when an analogy between cardinalities is found, the matching is created, with the corresponding coefficient. Only cardinalities of relations with the same arity are compared.

3.1.2 Using Domain Axioms

Domain Axioms are represented in MetaOCGL in order to compare their structures. For each axiom couple (a_1, a_2), where $a_1 \in O_1$ and $a_2 \in O_2$, the representations of a_1 and a_2 in MetaOCGL, $meta(a_1)$ and $meta(a_2)$, are built. These representations are automatically enriched by adding information about

Table 1: Modifications of the coefficient of the matching (r_1, r_2, c) according to the cardinalities of the relations. c_{min} and c_{max} are the values of cardinalities for the relations (for a given element of their signatures).

	Relation r_1 in O1	Relation r_2 in O2	Action
Min Card	0 (resp. $c_{min} \geqslant 1$)	$c_{min} \geqslant 1$ (resp. 0)	$-2 * W_{cmin}$
	$c_{min} \neq 0$	$c_{min} \neq 0$	$+2 * W_{cmin}$
	$c1_{min} \neq 0$	$c2_{min} \neq 0$ and $\neq c1_{min}$	$-W_{cmin}$
	Relation r_1 in O1	Relation r_2 in O2	Action
Max Card	∞ (resp. $c_{max} \geqslant 1$)	$c_{max} \geqslant 1$ (resp. ∞)	$-W_{cmax}$
	$c_{max} \neq \infty$	$c_{max} \neq \infty$	$+2 * W_{cmax}$
	$c1_{max} \neq \infty$	$c2_{max} \neq \infty$ and $\neq c1_{max}$	$-2 * W_{cmax}$

the nodes: for instance, in figure 3, the two relations *enemy* of the axiom *Enemy-Enemy* in OCGL are represented in MetaOCGL by the two concepts *Antecedent_R* which are linked by the meta-relation called *type_identity*, because the antecedent part of the Domain Axiom *Enemy-Enemy* in OCGL includes two instances of the same relation *Enemy*.

Two types of topological equivalence are then considered:

1. the **Equivalence**, that occurs when projections exist from $meta(a_1)$ to $meta(a_2)$ and from $meta(a_2)$ to $meta(a_1)$, without considering the $type\_identity$ relations;
2. the **Typed-Equivalence** that occurs when the two projections exist with the $type\_identity$ relations.

The weight of a typed-equivalence is higher than those of an equivalence. A typed-equivalence (resp. equivalence) between two axioms increases the coefficient of nodes linked by projection by the weight of the axiom typed-equivalence (resp. equivalence). When no projection (or only one) exists, no modification is done.

For example, the two Domain Axioms of figure 3 (*Enemy-Friend* and *Enemy-Enemy*) are equivalent because two projections exist between their meta-graphs without considering the *type-identity* relations. When considering the *type-identity* relations, there exists no projection, so they are not typed-equivalent.

4 RELATED WORK

Currently, a lot of tools that deal with finding correspondences between ontologies are proposed (Doan and Halevy, 2005; Noy, 2004; Shvaiko and Euzenat, 2005). The first way to classify these tools is to consider the objective which is pursued: (1) merging two ontologies to create a new one, (2) defining a transformation function that transforms one ontology into another or (3) defining a mapping between concepts or relations in two ontologies by finding pairs of related concepts/relations. Our work is dedicated to the latter objective. Note that although we are able to compare two axioms structurally, we have not yet considered the semantic mapping between axioms. Another way to categorize the tools is to consider the type of input on which the tool relies in its analysis and which it requires: (1) class names or natural-language definitions, (2) class hierarchy and properties, or (3) instances. Our approach is based on (2) and (4); we also introduces a new type of input: Axioms (including Schemata Axioms and Domain Axioms).

Then, in (Ehrig and Sure, 2004), a similarity stack is provided in order to classify the different measures that can be used to perform ontology matching. This stack is composed of five levels: the *Entities* level, the *Semantic Nets* level, the *Description Logics* level, the *Restrictions* level and the *Rules* level. For the first three levels, the authors provide similarity measures which of course differ according to semantic complexity of the level which is considered. However, for the *Restrictions* level and *Rules* level, no measure is proposed. Explanations given by the authors are the following: ”*the features like algebraic properties or equivalence/disjointness are not sufficiently used by the community to be considered as a material for similarity measure; for the Rules level, there has not been sufficient research and practical support for the Rule Layer of the Semantic Web Layer Cake*”. Our work must be considered as an extension of this classification in the sense that it provides measures based on the axioms of the domain which include both the *Restrictions* level and the *Rules* level. However, as we claim that it is not possible to consider rules and constraints at the ontological level (rules and constraints only exist at the operational level, we propose to modify the stack by merging the two levels *Restrictions* and *Rules* into only one: the *Axioms* level.

5 CONCLUSION

In this paper, we have introduced a new ontology matching approach. This approach, which mainly relies on graph-based representations and graph-based reasoning mechanisms, is particularly relevant to manage heavyweight ontologies since the main components of these ontologies are axioms which can be easily represented and compared with graph-based solutions. Our method has the advantage of incorporating most of the descriptive features of a heavyweight ontology into the matching process whereas most of the current methods cover only subsets of a lightweight ontology (mainly the hierarchy of concepts and their natural language expression). Of course, this method, although applicable, is not very efficient in a context of lightweight ontologies (and this is why we are not yet involved in the OAEI campaigns) . However, as demonstrated by the current challenge ”Reasoning the Semantic Web”, the need for developing heavyweight ontologies inevitably will increase in an immediate future. So, it seems interesting to focus on developing matching techniques dedicated to this type of ontology.

REFERENCES

Ashpole, B., Ehrig, M., Euzenat, J., and Stuckenschmidt, H. (2005). Ontology alignment evaluation initiative - 2005. In *Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies*. CEUR Proceedings - Volume 156.

Benjamins, R., Euzenat, J., Noy, N., Shvaiko, P., Stuckenschmidt, H., and Uschold, M. (2006). Ontology alignment evaluation initiative - 2006. In *Proceedings of the International Workshop on Ontology Matching*, http://oaei.ontologymatching.org/2006/.

Berners-Lee, T., Handler, J., and Lassila, O. (2001). The semantic web. In *Scientific American*, volume 248, pages 35–43.

Doan, A. and Halevy, A. (2005). Semantic integration research in the database community: A brief survey. In *AI Magazine, Special Issue on Semantic Integration*.

Ehrig, M. and Sure, Y. (2004). Ontology Mapping - an integrated approach. In *Proceedings of the First European Semantic Web Symposium*, pages 76–91. Springer-Verlag (LNCS 3053).

Fürst, F., Leclère, M., and Trichet, F. (2004). Operationalizing domain ontologies: a method and a tool. In de Mantaras, R. L. and Saitta, L., editors, *European Conference on Artificial Intelligence (ECAI'2004)*, pages 318–322. IOS Press.

Fürst, F. and Trichet, F. (2005a). Integrating domain ontologies into KBS. In *18th International Florida Artificial Intelligence Research Society Conference (FLAIRS'2005)*, pages 826–827. AAAI Press.

Fürst, F. and Trichet, F. (2005b). Toocom: bridge the gap between ontologies and knowledge-based systems. In *17th International Conference on Knowledge Engineering and Software Engineering (SEKE'2005)*, pages 235–243. KSI editors.

Gomez-Perez, A., Fernandez-Lopez, M., and Corcho, O. (2003). *Ontological Engineering*. Springer, Advanced Information and Knowledge Processing.

Noy, N. F. (2004). Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33(4):65–70.

Shvaiko, P. and Euzenat, J. (2005). A survey of schema-based matching approaches. In *Journal on Data Semantics (3730)*, pages 76–171. Springer-Verlag.

Sowa, J. (1984). *Conceptual Structures : information processing in mind and machine*. Addison-Wesley.