Ontology-driven Semantic Mapping

Domenico Beneventano[1], Nikolai Dahlem[2], Sabina El Haoum[2], Axel Hahn[2], Daniele Montanari[1,3], Matthias Reinelt[2]

[1] University of Modena and Reggio Emilia, Italy
{domenico.beneventano, daniele.montanari}@unimore.it
[2] University of Oldenburg, Germany
{dahlem, elhaoum, hahn, reinelt}@wi-ol.de
[3] Eni SpA, Italy
daniele.montanari@eni.it

Abstract. When facilitating interoperability at the data level one faces the problem that different data models are used as the basis for business formats. For example relational databases are based on the relational model, while XML Schema is basically a hierarchical model (with some extensions, like references). Our goal is to provide a syntax and a data model neutral format for the representation of business schemata.
We have developed a unified description of data models which is called the Logical Data Model (LDM) Ontology. It is a superset of the relational, hierarchical, network, object-oriented data models, which is represented as a graph consisting of nodes with labeled edges. For the representation of different relationships between the nodes in the data-model we introduced different types of edges. For example: *is_a* for the representation of the subclass relationship, *identifies* for the representation of unique key values, *contains* for the containment relationship, etc. In this paper we discuss the mapping process as it is proposed by EU project STASIS (FP6-2005-IST-5-034980). Then we describe the Logical Data-Model in detail and demonstrate its use by giving an example. Finally we discuss future research planned in this context in the STASIS project.

Keywords: business schema representation, business interoperability, meta-model

1. Introduction

Today's enterprises, no matter how big or small, have to meet the challenge of bringing together disparate systems and making their mission-critical applications collaborate seamlessly.

One of the most difficult problems in any integration effort is the missing interoperability at the data level. Frequently, the same concepts are embedded in different data models and represented differently. One difficulty is identifying and mapping differences in naming conventions, whilst coping with the problems of

polysemy (the existence of several meanings for a single word or phrase) and synonymy (the equivalence of meaning). A connected problem is identifying and mapping differences stemming from the use of different data models. For example information expressed in a relational schema is based on the relational data model, while XML Schema is basically a hierarchical model (with some extensions, like references).

Therefore we propose an ontology to describe a unified data model which is called the Logical Data Model. The purpose of the Logical Data Model ontology is to provide a common representation able to encapsulate substantial information coming from different sources and various schema formats. Data models represented by such an ontology can be considered as a neutral specification which allows a common processing in an interoperability framework.

In the remainder we first discuss the related work (section 2). Then we describe the mapping process to provide the context for this work in section 3. Section 4 presents the Logical Data Model Ontology and gives an example. Section 5 discusses ontology-driven semantic mapping and section 6 concludes with a discussion and an outlook to the future research.

2 Related Work

The integration costs for enterprise applications cooperation are still extremely high, because of different business processes, data organization, application interfaces that need to be reconciled, typically with great manual (and therefore error prone) intervention. This problem has been addressed independently by MDA and ontology-based approaches.

The Model Driven Architecture (MDA) proposed by the Object Management Group (OMG) uses platform-independent models (PIMs) [1] as the context for identifying relations between different applications. Transformation is a central concept in MDA to address how to convert one model into another model of the same system, and further into executable code. MDA provides technologies to handle meta models, constraints etc. which can be used for semantic enrichment and model transformation.

In model-based approach, Unified Modelling Language [2] is used to express conceptual models. The meta language Meta Object Facility (MOF) is defined as part of the solution in order to capture relationships between data elements. Transformation languages are used to create executable rules, and transformation techniques can be used in the process of detailing the information needed, converting from abstract MOF compliant languages to more formal ones [3].

Today, ontology technologies have reached a good level of maturity and their applications to industrial relevant problems are proliferating. Ontologies are the key elements of the Semantic Web. The notion of the Semantic Web is led by W3C and defined to be a “common framework allowing data to be shared and reused across application, enterprise and community boundaries” [4].

Ontologies support semantic mapping by providing explicitly defined meaning of the information to be exchanged. The development of the LDM Ontology was

particularly inspired by related work on relational schema modeling and the general goal of establishing mappings among (fragments of) domain ontologies. The latter has been an active field of research in the last ten years, exploring a number of approaches.

The basic expression of mapping for ontologies modeled with description logic formalisms and the associated languages (like OWL) involves the use of basic language constructs or evolved frameworks to express the existence and properties of similarities and then mappings [5] [6] [7]. One significant result in this area is the MAFRA framework [8]. Research in the area of database schema integration has been carried out since the beginning of the 1980s, and schema comparison techniques are often well suited for translation into mapping techniques. A survey of such techniques is offered in [9]. One system extensively using these techniques is MOMIS (Mediator Environment for Multiple Information Sources); MOMIS creates a global virtual view of information sources, independent of their location and heterogeneity [10].

The discovery of mappings has been studied by means of general methods often derived from other fields. One such approach is *graph comparison,* which comprises a class of techniques which represent the source and target ontologies (or schemas) as graph, and try to exploit graph structure properties to establish correspondences. Similarity flooding [11] and AnchorPrompt [12] are examples of such approaches.

Machine learning techniques have also been used. One such example is GLUE [13], where multiple learners look for correspondences among the taxonomies of two given ontologies, based on the joint probability distribution of the concepts involved and a probabilistic model for combination of results by different learners. Another example is OMEN (Ontology Mapping Enhancer) [14] which is a probabilistic mapping tool using a bayesian net to enhance the quality of the mappings.

Linguistic analysis is also quite relevant, as linguistic approaches exploit the names of the concepts and other natural language features to derive information about potential mates for a mapping definition. For example, in [15] a weighted combination of similarities of features in OWL concept definitions is used to define a metric between concepts. Other studies in this area include ONION [16] and Prompt [17], which use a combination of interactive specifications and heuristics to propose potential mappings. Similarly, [18] use a bayesian approach to find mappings between classes based on text documents classified as exemplars of these classes.

In Athena (ST-507849), a large European IST project, two different technologies have been applied to support model mapping. Semantic mapping involves the application of an ontology. However, current literature does not provide detailed description regarding how this is to be done, as pointed out by [19] and [20]. In Athena, a solution has been proposed, based on semantic annotation (A* tool), reconciliation rules generation (Argos tool), and a reconciliation execution engine (Ares). Parallel, in Athena, also a model-based approach has been proposed, based on a graphic tool (Semaphore) aimed at supporting the user in specifying the mappings and XSLT based transformation rules.

Other European projects addressing mapping issues include the IST FP6 projects SWAP (IST-2001-34103) [21], SEKT (IST-2003-506826) [22], and DotKom (IST-2001-34038) [23].

3 Mapping of Business Schemata

When analyzing semantic relations between business schemata, we follow the approach of A* [24] to obtain a neutral representation of the schemata first. In a subsequent step this neutral representation is processed to identify mappings. These steps are discussed in the following two sections.

3.1 Obtaining a neutral schema representation

The proposed mapping process works on a neutral representation, which abstracts from the specific syntax and data model of a particular business schema definition. Therefore, all incoming business schemata first need to be expressed in this neutral format. Fig. 1 shows the steps of this acquisition process.

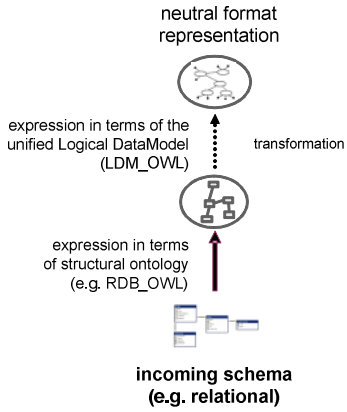

Fig. 1. Schema acquisition process

Firstly, the incoming schema is expressed in terms of a corresponding structural ontology. Several parseable and non-parseable schema formats are already analyzed and supported namely relational databases, XML schema, EDIFACT-like EDI environments, FlatFile representations. For each of these formats a specific structural ontology is defined [25].

Then, in a second step, the model specific structural ontology representation is transformed into a neutral representation which is based on the Logical Data Model. This transformation can be automated by applying a set of predefined rules.

3.2 Identification of mappings

Once the schema information has been acquired and expressed in the unified model, further analysis and/or processing can be performed to identify a set of mappings between semantic entities being used in different business schemata. The

goal is to provide such sets of mappings as input to translator tools to achieve interoperability between dispersed systems without modifying the involved schemata.

The definition of the mappings is done through the acquisition of the crucial features in the schemata of the source and target, giving them a conceptual representation, enriching this representation with semantic annotations and then using the system functionalities to synthesize and refine the mappings. An overview of this process is given in Fig. 2.

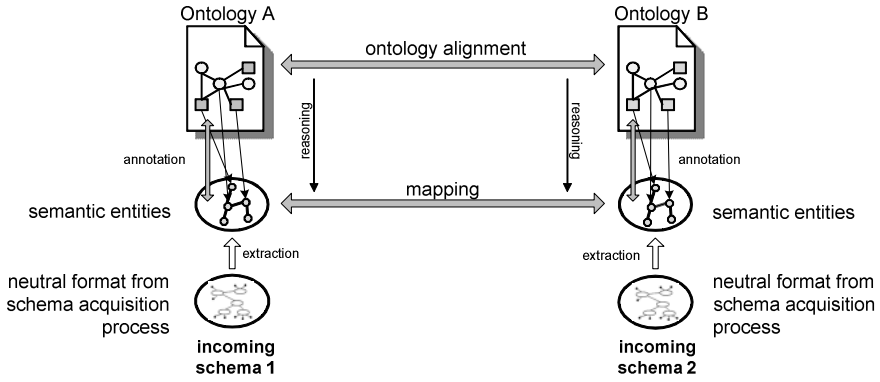

Fig. 2. Mapping process

As shown in Fig. 2 the neutral representation of incoming schemata provides the basis for the identification of the relevant semantic entities being the basis of the mapping process. This step is labeled “extraction” and the resulting semantic entities are the a-box of the LDM Ontology. Apart from the element being identified as semantic entity, a semantic entity holds metadata such as annotations, example values, keywords, owner and access information, etc. The analysis of the information encapsulated in semantic entities can support the finding of mapping candidates.

A more advanced way to identify mappings between semantic entities is to derive them through reasoning on aligned ontologies. For this purpose the semantic entities need to be annotated with respect to some ontology as proposed in A*. Based on the annotation made with respect to the ontologies and on the logic relations identified between these ontologies, reasoning can identify correspondences on the semantic entity level and support the mapping process. Beyond the capability of A* this reasoning can also benefit from the conceptual information derived from the LDM Ontology because all semantic entities carry this extra information by being instances of the concepts of the LDM Ontology.

4 Logical Data Model Ontology

This section contains a general description of the Logical Data Model Ontology followed by an example to demonstrate its main characteristics.

4.1 General description of the model

The LDM Ontology contains generic concepts abstracting from syntactical aspects and different data models. As an intuitive example, in the relational model a foreign key expresses a reference between two tables; at a more abstract level we can consider the two tables as nodes of a graph and the foreign key as an edge from one table to another table; more precisely this is a directed (since a foreign key has a "direction") and labeled (since we want to distinguish two foreign keys between the same pair of tables) edge.

In this way, the LDM Ontology corresponds to a graph with directed labeled edges and it has the following types of concepts:

1. The *Nodes* of the graph, which are partitioned in *SimpleNodes* and *ComplexNodes*.
2. The edges of the graph, which represent *Relationships* between *Nodes.* The following types of *Relationships* can exists:
   - *Reference*: A *Reference* is a directed labeled edge between *ComplexNodes.*
   - *Identification*: A *ComplexNode* can be identified by a *SimpleNode* or a set of *SimpleNodes.*
   - *Containment*: A *ComplexNode* can contain other *Nodes*, *SimpleNodes* and/or *ComplexNodes.*
   - *Qualification*: A *Node* can be qualified by a *SimpleNode.*
   - *Inheritance*: *Inheritance* can exist between *ComplexNodes*.

The LDM Ontology has been represented as an OWL ontology. An overview of the concepts and their relations in the ontology is shown in Fig. 3. A detailed description of the LDM Ontology is provided by [25]

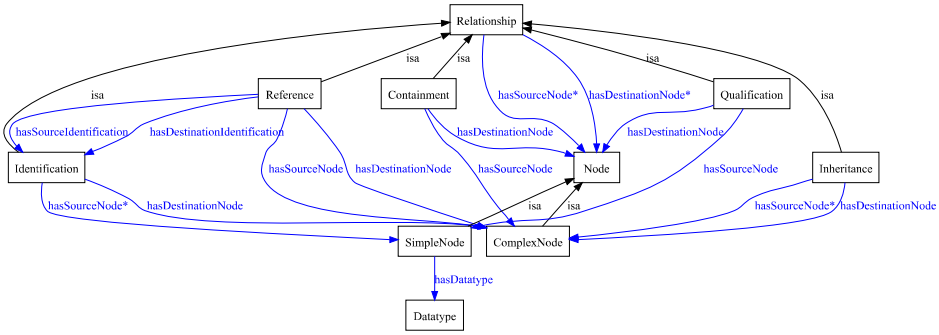

Fig. 3. Overview of the concepts in the LDM Ontology

4.2 Demonstration example

In this section an example is introduced to show how a relational data base schema is first represented in terms of a structural ontology and then transformed into a LDM Ontology representation by means of respective transformation rules.

For the relational case the structural ontology has to provide concepts for the terms *Database*, *Relation* and *Attribute* and a property *consistsOf* to create a hierarchy involving them. For this purpose the structural ontology contains the concepts of *Catalogue*, *Table* and *Column* and the object property *hasColumn.*

Consider the relational schema in Fig. 4. Expressed in terms of the structural ontology for the relational case (hereafter shortly referred as relational structural schema) there are two *Tables*: *Table* "Order" and *Table* "OrderLine" with their *Columns* "number", "date", "customerID" and "articleNumber", "quantity", "lineNumber", "orderNumber" respectively. Additionally, the *Column* "number" is declared a *PrimaryKey* of the "Order" *Table* and the *Column* "lineNumber" the *PrimaryKey* of the *Table* "OrderLine". Further, the "OrderLine" is connected to one specific "Order" using a *ForeignKey* reference "FK_OrderLine_Order".

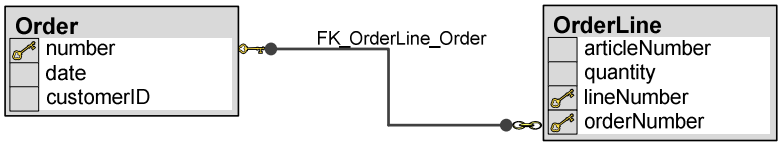

Fig. 4. Part of an exemplary relational data base schema

The next step to achieve an LDM Ontology representation is to apply transformation rules to the structural ontology representation. A brief overview of the transformation rules is presented in Table 1. Due to space limitation the table only gives an intuition of the rules; their detailed explanation is given in [25]. In general, in the LDM Ontology all *Tables* will be represented as *ComplexNodes, Columns* as *SimpleNodes*, and so on.

Table 1. Transformation rules from the relational structural ontology into a LDM Ontology representation

Entity in the relational structural ontology	Entity in the Logical Data Model	Comments
Table	*ComplexNode*	All *Tables* are represented as *ComplexNodes*
Column	*SimpleNode*	All *Columns* are represented as *SimpleNodes*
KeyConstraint	*Identification*	All *KeyConstraints* (i.e. *PrimaryKeys* and *AlternativeKeys*) are represented as *Identifications*
ForeignKey	*Reference*	All *ForeignKeys* are represented as *References*
hasColumns	*Containment*	The relationship between a *Table* and its *Columns* is represented as a *Containment* relationship

The application of the transformation rules leads to an LDM Ontology based representation of the example as shown in Fig. 5. The notation used in this figure is described by [25].

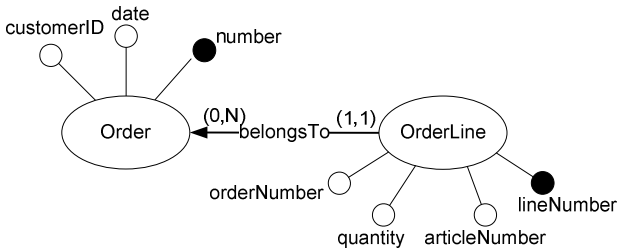

Fig. 5. LDM Ontology representation of the exemplary schema

According to the graphical representation in Fig. 5 the example schema contains two *ComplexNodes* "Order" and "OrderLine". For each *Column* a *SimpleNode* is introduced and connected with its *Table*/*ComplexNode* via the *Containment* relation. *Identification* relations are defined for the *PrimaryKeys* "number" and "lineNumber". The *ForeignKey* "FK_OrderLine_Order" is transformed to a *Reference* "belongsTo".

5. Ontology-driven Semantic Mapping

As discussed in the section 3 mappings between Semantic Entities can be achieved based on annotations linking the semantic entities with some concepts being part of an ontology.

The annotation of semantic entities with respect to external ontology means that additional machine processable knowledge is associated with them. As in A* the ontology-driven process of deriving correspondences between semantic entities belonging to different schemata will make use of this additional knowledge. Our approach also benefits from structural knowledge on the data model represented by linking the semantic entities to the concepts of the LDM Ontology.

When the annotation of semantic entities belonging to different schemata is based on one common ontology and the LDM Ontology (see Fig. 6), the annotations can directly facilitate the discovery of semantic relations between the semantic entities.

The definition of semantic link specification (SLS) is based on [26]. The following semantic relations between semantic entities of two business formats are defined: equivalence (EQUIV); more general (SUP); less general (SUB); disjointness (DISJ).

As in [26], when none of the relations holds, the special IDK (I do not know) relation is returned; Notice IDK is an explicit statement that the system is unable to compute any of the declared (four) relations. This should be interpreted as either there is not enough background knowledge, and therefore, the system cannot explicitly compute any of the declared relations or, indeed, none of those relations hold according to an application. The semantics of the above relations are the obvious set-theoretic semantics.

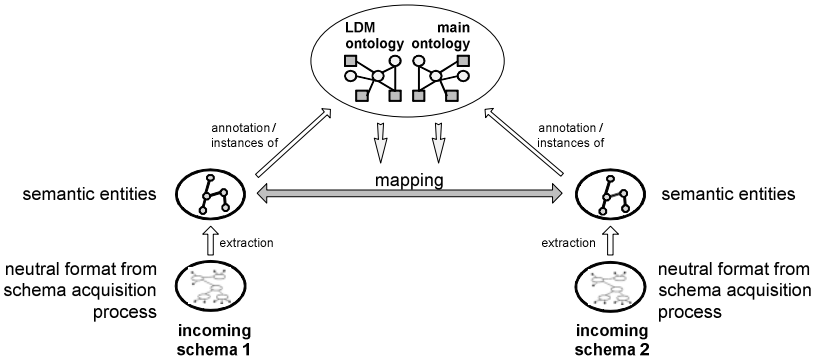

Fig. 6. Ontology-based schema mapping with a single common ontology

More formally, an SLS is a 4-tuple <ID, semantic_entity1, R, semantic_entity2> where ID is a unique identifier of the given mapping element; semantic_entity1 is an entity of the first format; R specifies the semantic relation which may hold between semantic_entity1 and semantic_entity2; semantic_entity2 is an entity of the second format.

Our discussion is based on examples. To this end we consider the following two business formats: The graphical representation of semantic_entity1 from a business format 1 (bf1) shown in Fig. 7 and semantic_entity2 from another business format 2 (bf2) shown in Fig. 8. We consider the annotation of the above business format with respect to the Purchase_order Ontology (see Fig. 9).

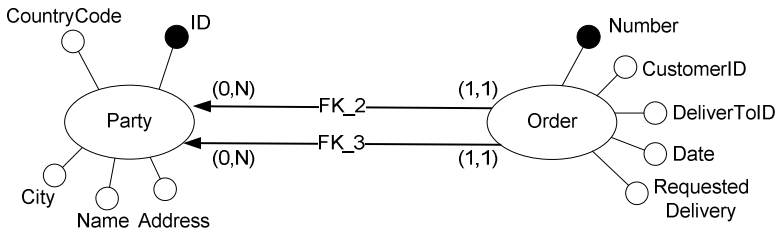

Fig. 7. Business format specification (bf1) (derived from relational schema)

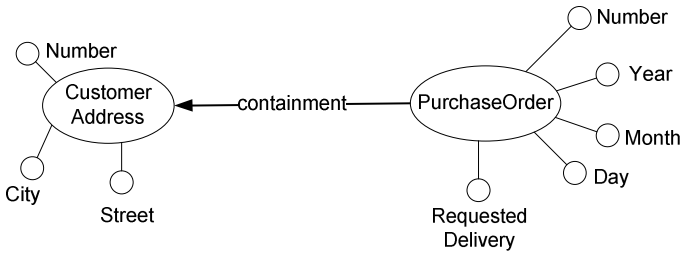

Fig. 8. Business format specification (bf2) (derived from an XML schema)

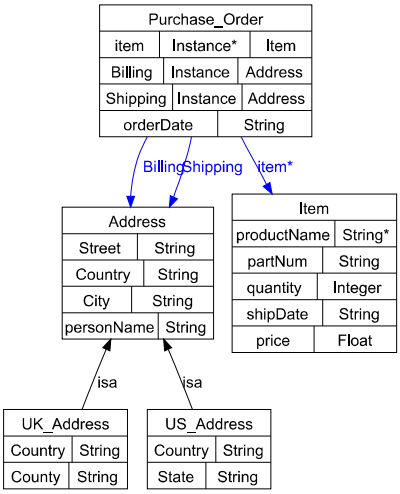

Fig. 9. The ontology of Purchase_order

The proposed "Ontology-based schema mapping with a single common ontology" is based on the annotation of a business format with respect to this single common ontology.

Here we will use the following proposal. An annotation element is a 4-tuple <ID, SE, R, concept> where ID is a unique identifier of the given annotation element; SE is a semantic entity of the business format; concept is a concept of the ontology; R specifies the semantic relation which may hold between SE and concept.

The proposal is to use the following semantic relations between semantic entities of the business format and the concepts of the ontology: equivalence (AR_EQUIV); more general (AR_SUP); less general (AR_SUB); disjointness (AR_DISJ).

Let us give some examples of annotation. In the examples, the unique identifier ID is omitted.

- (bf2:Address, AR_EQUIV,O:Address) may be considered as the output of automatic annotation
- (bf2:Address, AR_SUB,O:Address) may be considered as the output of a ranked automatic annotation/search: the AR_SUB relation instead of AR_EQUIV is used since the rank is less than a given threshold
- (bf2:Address, AR_EQUIV, O:Address and Billing-1.Purchase_Order) may be considered as a refinement by the user of (bf2:Address, AR_EQUIV,O:Address) to state that the address in the BF is exactly the "address of the Billing in a Purchase_Order"

Let us also consider the following possible annotations of bf1

- (bf1:Address, AR_EQUIV,O:Address)
- (bf1:Address, AR_SUB,O:Address)
- (bf1:Address, AR_EQUIV, O:Address and Shipping-1.Purchase_Order)
- (bf1:Address, AR_EQUIV, O:Address and Shipping-1.Purchase_Order)
- (bf1:Address, AR_DISJ, O:Address and Billing-1.Purchase_Order)

Now, some example of the SLS derived from annotation will be discussed. To this end, let us suppose that in the bf2 there is the following annotation for address (bf2:Address, AR_EQUIV,O:Address).

We want to discuss what is the SLS derived between bf2:Address and bf1:Address, by considering the following cases for the Address annotation in bf1

Case 1) (bf1:Address, AR_EQUIV,O:Address)

The following SLS can be derived (bf1:Address, EQUIV, bf2:Address)

Case 2) (bf1:Address, AR_SUB,O:Address) and

The following SLS can be derived (bf1:Address, SUB, bf2:Address)

Case 3) (bf1:Address, AR_EQUIV, O:Address and InverseOf(Shipping)-.Purchase_Order)

The following SLS can be derived (bf1:Address, SUB, bf2:Address) since Address and InverseOf(Shipping).Purchase_Order (the annotation of bf1:Address) is subsumed by Address (the annotation of bf2:Address).

This shows how the semantic mapping can be derived from the semantic entity specification. The information of the linkage to the LDM Ontology is used in the same way. One topic is still open. A possible extension [to be evaluated] w.r.t. the [26] framework is the addition of the overlapping (OVERLAP) semantic relation. Formally, we need to evaluate if with OVERLAP we can decide IDK relations; moreover we need to proof that with OVERLAP "relations are ordered according to decreasing binding strength"

6. Discussion and Future Research

We provide a joint approach to integrate the benefits of the MOF and ontology based semantic mapping methods. Model entities of business formats/standards are described by a generic meta model which is made explicit by an ontology, called the Logical Data Model Ontology. By annotating these semantic entities w.r.t. business ontologies an enriched knowledge base is available to reason on semantic links to align the entities of business formats. These technologies are going to be integrated in an interoperability framework to share the semantic information in peer groups to enrich the semantic basis for cooperation. This enhances the common ontology to provide an even better basis for the mapping process. This is accompanied by further approaches to simplify the definition of ontologies, their linkage to semantic entities (annotation) and verification of the jointly generated semantic net.

Acknowledgments. The STASIS IST project (FP6-2005-IST-5-034980) is sponsored under the EC 6th Framework Programme.

References

[1] Kleppe, A., Warmer, J., Bast, W.: MDA Explained: The Model Driven Architecture--Practice and Promise. Addison-Wesley, Boston (2003)

[2] Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual. Second Edition, Addison-Wesley, Boston (2005)

[3] Karagiannis, D., Kühn. H.: Metamodelling Plattforms. In Bauknecht K., Min Tjoa, A., Quirchmayr, G. (eds.): Thirst Int. Conference EC-Web 2002, p. 182. Springer, Berlin (2002)
[4] W3C Semantic Web Activity, http://www.w3.org/2001/sw/ last accessed 2007-10-24
[5] Ehrig, M., Haase, P., Hefke, M., and Stojanovic, N.: Similarity for Ontologies – a Comprehensive Framework. In Workshop Enterprise Modelling and Ontology: Ingredients for Interoperability, PAKM 2004 (2004)
[6] Weinstein, P., Birmingham, W.P.: Comparing Concepts in Differentiated Ontologies. In 12th Workshop on Knowledge Acquisition, Modelling, and Management (KAW99), Banff, Alberta, Canada (1999).
[7] Choi, N., Song, I-Y., Han, H.: A Survey of Ontology Mapping. In SIGMOD Record 35, Nr. 3 (2006)
[8] Mädche, A., Motik, B., Silva, N., Volz, R: MAFRA - A Mapping Framework for Distributed Ontologies. In 13th Int. Conf. on Knowledge Engineering and Knowledge Management, vol. 2473 in LNCS, pp. 235--250 (2002)
[9] Rahm, E., Bernstein, P.: Survey of Approaches to Automatic Schema Matching. VLDB Journal 10 (2001)
[10] Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.: Synthesizing an Integrated Ontology, IEEE Internet Computing, September-October (2003)
[11] Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm and its Applications to Schema Matching. In 18th International Conference on Data Engineering (ICDE-2002), San Jose, California, (2002)
[12] Noy, N.F., and Musen, M.A.: Anchor-PROMPT: Using non-local context for semantic matching. In Workshop on Ontologies and Information Sharing at the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA, US (2001)
[13] Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In The 11th Int. WWW Conference, Hawaii, USA (2002)
[14] Mitra, P., Noy, N.F., Jawal, A.R.: OMEN: A Probabilistic Ontology Mapping Tool. LNCS, Vol. 3729/2005, Springer, Berlin (2005)
[15] Euzenat, J., Valtchev, P.: Similarity-based ontology alignment in OWL-Lite. In The 16th European Conference on Artificial Intelligence (ECAI-04), Valencia, Spain (2004)
[16] Mitra, P., Wiederhold, G., Decker, S.: A scalable framework for interoperation of information sources. In SWWS01, Stanford University, Stanford, CA, US (2001)
[17] Noy, N.F., Musen, M.A.: The PROMPT suite: Interactive tools for ontology merging and mapping. International Journal of Human-Computer Studies, 59, pp. 983--1024 (2003)
[18] Prasad, S., Peng, Y., Finin, T.: A tool for mapping between two ontologies using explicit information. In AAMAS 2002 Ws on Ontologies and Agent Systems, Bologna, Italy (2002)
[19] Wache, H, Vögele T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann H., Hübner, S.: Ontology-Based Integration of Information - A Survey of Existing Approaches. In: IJCAI-01 Workshop Ontologies and Information Sharing (IJCAI-01), pp. 108-118 (2001)
[20] Gómez-Pérez, A., Fernández-López M, Corcho, 0.: Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. Springer (2004)
[21] SWAP Project Web Site, http://swap.semanticweb.org/ last accessed 2007-10-24
[22] SEKT Project Web Site, http://www.sekt-project.com/ last accessed 2007-10-24
[23] DotKom Project Web Site, http://nlp.shef.ac.uk/dot.kom/ last accessed 2007-10-24
[24] Callegari, G. Missikoff, M., Osimi, N., Taglino F.: Semantic Annotation language and tool for Information and Business Processes Appendix F: User Manual, ATHENA Project Deliverable D.A3.3 (2006) available at http://leks-pub.iasi.cnr.it/Astar/AstarUserManual1.0 last accessed 2007-10-24
[25] STASIS Project Web Site, http://www.stasis-project.net/ last accessed 2007-10-24
[26] Giunchiglia, F., Yatskevich, M., Shvaiko. P.: Semantic Matching: Algorithms and Implementation. Journal on Data Semantics (JoDS), IX, LNCS 4601, pp. 1-38 (2007)