An Instance Mapping Ontology for the Semantic Web

Yuangui Lei
Knowledge Media Institute
The Open University
Walton Hall, Milton Keynes, MK7 6AA
y.lei@open.ac.uk

ABSTRACT
Semantic data transformation plays an important role in realizing the vision of the semantic web. It supports the transformation of data in different representations into ontologies. In order to allow the task to be achieved effectively, the instructions on how to realize transformation should be well specified, preferably in a declarative and reusable format, thus allowing the construction of robust tools which on the one hand assist users to generate and maintain mappings at design time and on the other hand perform semantic data transformation at run time. Furthermore, the transformation instructions should not only allow the generation of semantic data objects but also allow the creation of rich semantic relations between them. In this context, we developed a comprehensive instance mapping ontology. One distinctive feature of the instance mapping ontology is that it provides comprehensive support for the specification of complex mappings. Another feature is that the instance mapping ontology is representation independent, which does not limit itself to data sources in particular representations. This ontology has been applied in generating a semantic layer for the web site of the Knowledge Media Institute (KMi) at the Open University.

Categories and Subject Descriptors
I.2.4 Knowledge Representation Formalisms and Methods – *representation languages, semantic networks.*

General Terms
Management, Languages

Keywords
Instance mapping, ontology mapping, mapping ontology, semantic data transformation, semantic web

INTRODUCTION
The semantic web [1] is a vision of the next generation of the World Wide Web in which information is given well-defined meaning understandable by machines as well as humans. Even though there is still a way to go before this vision is fulfilled, adding a semantic data layer to ordinary web sites is an important step in the right direction. Sophisticated services, such as semantic searching and semantic indexing, can then be defined to provide easier access to the underlying knowledge for both end users as well as applications. Hence, it is highly desirable that knowledge can be extracted, verified, and integrated from the underlying heterogeneous sources. To achieve this goal, a crucial task is to extract data from different data sources, which is to transform data from different representations into the specified ontologies. This requires well specified transformation instructions.

Firstly, the transformation instructions should be specified in a declarative format, thus allowing the construction of robust tools which on the one hand assist users to generate and maintain mappings at design time by incorporating automatic or semi-automatic mapping discovery technologies [2, 10] and on the other hand execute semantic data transformation at run time by reasoning upon the mappings. Secondly, the transformation instructions should not only allow the generation of semantic data objects, but also allow the creation of rich semantic relations between them. This enables the generation of rich knowledge networks in the target ontology. Thirdly, the transformation instructions should be representation language independent. Thus, a data transformation engine can work on data in different representation languages with minimum efforts of configuration.

In the field of ontology mapping, a number of approaches have been developed, which provide languages for specifying mappings between ontologies [5, 9, 4, 6, 7]. Among them, MAFRA [7] is the only approach which provides an explicit ontology to support declarative specification of mappings. The MAFRA mapping ontology provides comprehensive support for the specification of mappings between classes and slots in terms of *ConceptBridge* and *AttributeBridge*. However, one important mapping relation is missing, which is the mapping that allows the generation of relations between semantic data objects. Without this capability, the MAFRA mapping ontology can only produce semantic data objects with few semantic relations.

In this work, we have developed a comprehensive instance mapping ontology with special focuses on i) the support for the generation of semantic data objects with rich semantic

K-CAP'05, October 2–5, 2005, Banff, Alberta, Canada.

relations and ii) the support for data transformation in a representation language independent manner.

The rest of the paper is organized as follows. We begin by presenting an overview of the instance mapping ontology. We then present an application of the instance mapping ontology, which transforms data in heterogeneous representations into formalized knowledge for the web site of the Knowledge Media Institute (KMi)[1] at the UK's Open University. In particular, we present a number of examples which illustrate the full specification support provided by the instance mapping ontology. Thereafter, we describe the related work and re-iterate the major contributions of this work.

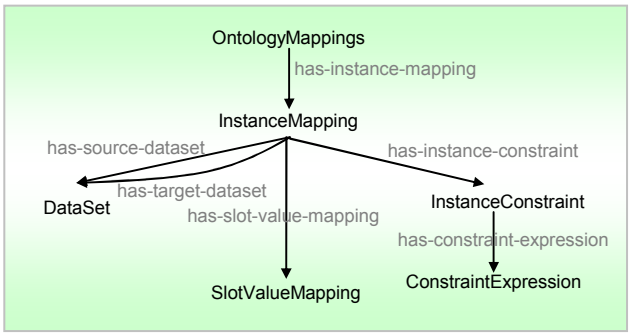

Figure 1. An overview of the mapping ontology

AN OVERVIEW OF THE INSTANCE MAPPING ONTOLOGY

Figure 1 shows an overview of the instance mapping ontology whose constructs are described in table 1. The instance mapping ontology relies on the following major constructs to support the specification of mappings between source schemas and the specified ontology:

- *Ontology Mappings*, which groups the specified instance mappings together to allow the generation of semantic data from different data sources.
- *Instance Mapping*, which expresses how to generate instances for the associated target class from the source data objects of the specified source class (or database table).
- *Instance Constraint*, which describes constraints on the data objects of the specified source class.
- *SlotValueMapping*, which specifies how to generate values for the specified slots.
- *DataSet*, which describes how to access data sets that hold source data objects or place the transformed semantic data objects.
- *ConstraintExpression*, which formulates constraints

Table 1. Constructs of the instance mapping ontology

Construct	Description	Slots
OntologyMappings	Grouping the mapping instructions for generating semantic data entries.	• has-instance-mapping
InstanceMapping	Describing how to generate instances from the specified source data objects	• has-source-dataset • has-target-dataset • has-source-class-name • has-target-class-name • has-instance-constraint • has-slot-value-mapping • has-correspondent-slot
InstanceConstraint	Describing constraints on the instances of the specified source class	• has-constrained-source-dataset • has-constrained-class-name • has-constrained-expression
SlotValueMapping	Expressing how to generate values for the specified target slot from the specified sources	• has-target-slot • values-from-slot • values-from-existing-instances • values-from-instances-to-be-created
DataSet	Describing how to access databases, knowledge bases, or XML documents.	• has-db-name • has-db-url • has-db-server
ConstraintExpression	Expressing constraints	• has-constrained-slot-name • has-constrained-operator • has-constrained-value • has-logic-operator

In the following sub sections, we will briefly describe the support that the instance mapping ontology provides for semantic data transformation, including i) the generation of semantic data objects, ii) the generation of semantic relations, and iii) the representation language independent data transformation. Such support will be illustrated in the next section.

Support for the generation of semantic data objects

Figure 2 illustrates the mechanism of semantic data transformation supported by the instance mapping ontology. Firstly, source data objects are extracted from the specified source data set. Secondly, constraints are specified on source data objects, which filter transformable data objects. Thirdly, transformation instructions are defined which describe how to generate slot values for instances of the target class. Finally, the transformed semantic data objects are stored in the specified target data set.

[1] http://kmi.open.ac.uk

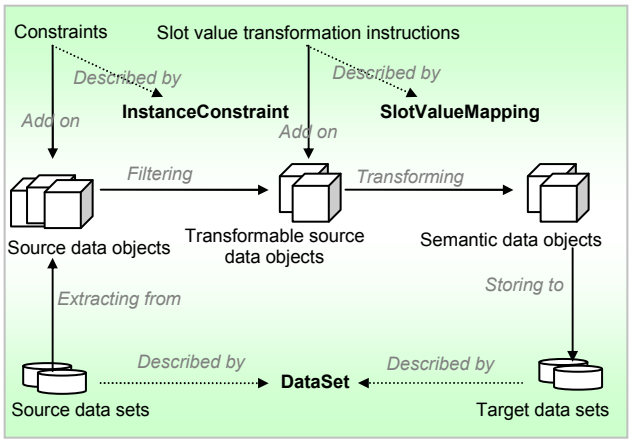

Figure 2. The mechanism of semantic data transformation

Source data sets and target data sets are specified in terms of the construct *DataSet*. Constraints are expressed by the construct *InstanceConstraint*. Slot value mappings are described by the construct *SlotValueMapping*. Thus, as will be illustrated later in the next section, an instance mapping specification comprises four major parts: i) the source data set and the source class whose instances are to be extracted as source data objects, ii) the target data set and the target class, iii) constraints on the source data objects, and iv) slot value transformation instructions which express how to generate values for the target semantic data objects (i.e. instances of the target class).

Support for the generation of relations between semantic data objects

Unlike the MAFRA mapping ontology, which only supports a single way of mapping values for slots as discussed earlier, our instance mapping ontology supports three ways of generating values: i) from the specified slot of the source data, ii) from existing instances which meet certain constraints, and iii) from instances which are to be generated in the mapping process. In particular, the latter two ways support the generation of references between semantic entities. This is supported by the construct *SlotValueMapping*.

Firstly, the construct *SlotValueMapping* relies on the slot *values-from-existing-instances* to specify the instances which will be referenced as values of the target slot in forms of instance constraints. Secondly, the construct *SlotValueMapping* makes use of the slot *values-from-instances-to-be-created* to express the situation when values of the target slot need to be mapped from instances which in turn need to be generated in data transformation. The data type of the slot *values-from-instances-to-be-created* is the class *InstanceMapping*, which means that a sub mapping is invoked when generating values for the specified target slot. More specifically, the target instance of the sub mapping is referenced as the values of the target slot. On the other hand, as will be illustrated later, the sub mapping relies on the slot *has-correspondent-slot* to reference the instance that invokes this sub mapping. Thus, semantic relations are established between the target instances of the sub mapping and those of the mapping which invokes the sub mapping.

Support for representation language independent data transformation

The instance mapping ontology relies on the construct *DataSet* to describe how to access data sets involved in data transformation (including source data sets and target data sets) in terms of *has-db-url*, *has-db-server*, and *has-db-name*. The slot *has-db-url* describes the location of the data set; the slot *has-db-server* indicates the database server (e.g., WebOnto [3] and MySQL[2]) which supports the management of the data set, and the slot *has-db-name* expresses the name of the data set. Thus, a data transformation engine is able to access, retrieve, and update data with the support of functionalities provided by the specified database or knowledgebase management system. In this way, the instance mapping ontology supports data transformation in a representation language independent way. However, in the cases when there is no management system available for accessing the specified data set, the data transformation engine will have to provide such services.

AN APPLICATION

The KMi web site is a non-semantic web site, which presents information about our lab. The underlying data sources are stored in departmental databases, knowledge bases, and HTML pages. Specifically, the information about people, technologies and projects is maintained in our departmental databases. Information about publications is stored in an internal knowledge base which also includes other information about activities of lab members, such as presentations given. In addition, the domain data also comprise an archive[3] of several hundred news items, describing events of significance to the KMi members. In order to provide a deep insight into the events, information such as *persons*, *organizations*, and *projects* needs to be pulled out and made accessible.

In this work, we took the KMi web site as a study case and applied the instance mapping ontology to specify the mapping instructions which allow the transformation of the underlying data of the KMi web site into formalized knowledge in terms of the specified domain ontology. Specifically, three major data sources need to be transformed, which include i) the XML [15] mark-ups produced by the ESpotter [11] service when marking-up KMi news stories, ii) records of projects and persons stored in departmental databases, and iii) publications represented in another ontology called kmi-impact-ontology. The domain ontology is

[2] http://www.mysql.com/ (Accessed 5 May 2005)
[3] http://news.kmi.open.ac.uk/rostra/news.php?r=6&t=1 (Accessed 5 May 2005)

built on the top of the *AKT reference ontology*[4], which abstracts information about personnel, projects, technologies, events, etc.

The transformed knowledge of the KMi web site can be accessed at http://semanticweb.kmi.open.ac.uk. A range of sophisticated semantic search services have been defined, which provide an easier access to domain knowledge. Examples include a semantic keyword searching service and a semantic question answering service, which are accessible from the above URL.

In this section, we illustrate the instance mapping ontology by two examples. The first example is simple but shows the specification skeleton of the instance mapping ontology. The second one is more complex, which shows how our instance mapping ontology provides comprehensive support for the specification of complex mappings.

Transforming data from XML sources

The following XML code shows a fragment of the result of the ESpotter service when marking up the news story titled as *First Summer School on Ontological Engineering and the Semantic Web*[5].

```
<Planet-News-Page id="251">
  <has-title>First Summer School on Ontological
Engineering and the Semantic Web </has-title>
  <mentions-organization>
      <instance>EPSRC</instance>
      …
  </mentions-organization>
  <mentions-person>
      <instance>Enrico Motta</instance>
      …
  </mentions-person>
   …
</Planet-News-Page>
```

The target class is the class *kmi-planet-news-item*. The following OCML [8] code shows a fragment of the definition of this class. As we can see from the definition of the target class and the XML schema implied in the XML data object, the mapping is fairly straightforward. Each source data object needs to be mapped to an instance of the target class without any constraint. The values of most target slots come from the values of the attributes of the entity *Planet-News-Page*.

```
(def-class kmi-planet-news-item (news-item)
  ((mentions-person :type person)
   (mentions-technology :type technology)
   (mentions-organization :type organization)
   (mentions-project :type project)
    …))
```

The following OWL [13] code shows a skeleton of the mapping specification. The specification comprises three major parts. The first part defines source data sets and specifies how to access them. The second part specifies the target data set. The third part includes specifications of how to generate instances from source data objects.

```
   <!-- part I: specifying source data sets -->
<mo:DataSet rdf:ID="espotter-markup-document">
   <mo:has-db-url>file:/d:/espotter-
result/notification.xml</mo:has-db-url>
   <mo:has-db-server>XML</mo:has-db-type>
   <mo:has-db-name>notification.xml</mo:has-db-
name>
   …
</mo:DataSet>

  <!-- part II: specifying the target data set -->
<mo:DataSet rdf:ID="kmi-basic-portal-kb" >
  <mo:has-db-url>plainmoor.open.ac.uk:3000
  </mo:has-db-url>
  <mo:has-db-server>webonto</mo:has-db-type>
  <mo:has-db-name>kmi-basic-portal-kb</mo:has-db-
name>
  …
</mo:DataSet>

<!-- part III: specifying instance mappings -->
<mo:InstanceMapping>
…
</mo:InstanceMapping>
…
```

As mentioned earlier, the specification of an instance mapping comprises four major parts. The first part specifies source data objects. The second part describes the target data set and the target class entity. The third part expresses constraints, which in this example is empty, as each source data object is mapped to the target ontology. The fourth part describes how to generate values for the target data objects from source data. The following code shows a fragment of the specification of this instance mapping example.

```
<mo:InstanceMapping rdf:ID="instance-mapping1" >
 <!-- part I: the source data objects
  -->
 <mo:has-source-dataset
    rdf:resource="#espotter-markup-document" />
  <mo:has-source-class-name>Planet-News-Page
  </mo:has-source-class-name>

<!-- part II: the target data set and the target
class entity -->
  <mo:has-target-dataset
       rdf:resource="#kmi-basic-portal-kb"/>
  <mo:has-target-class-name>kmi-planet-news-item
  </mo:has-target-class-name>

<!-- part III: constraints. In this example, it is
empty -->

<!-- part IV: slot value transformation instruc-
tions -->
  <mo:has-slot-value-mapping
    rdf:resource="#mentions-person-mapping"/>
</mo:InstanceMapping>
…
<mo:SlotValueMapping
   rdf:ID="mentions-person-mapping">
  <mo:has-target-slot>mentions-person
  </mo:has-target-slot>
  <mo:values-from-slot>mentions-person
  </mo:values-from-slot>
</mo:SlotValueMapping>
```

[4] http://kmi.open.ac.uk/projects/akt/ref-onto/ (Accessed 5 May 2005)
[5] http://news.kmi.open.ac.uk/rostra/news.php?r=11&t=2&id=446 (Accessed 5 May 2005)

Transforming data from sources in different ontologies

As mentioned earlier, in the KMi web site, information of publications is maintained in an internal knowledge base called *KMi-impact-kb* in terms of the ontology *kmi-impact-ontology*. In particular, all publications are represented by means of the class *Publications*. The domain ontology of the KMi web site on the other hand provides a very different vocabulary. First, it classifies publications into different classes, such as *article-in-a-journal*, *paper-in-proceedings*, *book-section*, *conference-proceedings*, *workshop-proceedings*, etc. Hence, publications in the source ontology need to be transformed into instances of different classes according to their specific features. For example, a conference paper needs to be mapped to an instance of the class *paper-in-proceedings*, while a journal paper needs to be mapped to an instance of the class *article-in-a-journal*. Second, there are a number of classes in the target ontology which have semantic relations with these classes. They form rich knowledge networks together. Figure 3 shows an example network which is formed by a conference paper.

Figure 4 shows the mappings involved when mapping a conference paper from the source ontology to the target ontology. The mapping *im1* is the main instance mapping, which describes how to transform a conference paper instance into an instance of the class *paper-in-proceedings*. The transformable source instances must be conference papers. Hence, there is a constraint defined in this mapping, constraining values of the slot *conference-name* of the source instances not to be empty. Other mappings shown in figure 4 are sub ones which are invoked when generating values for those slots whose values come from instances that need to be generated through the specified further instance mappings. Table 2 describes the major mappings involved in transforming a conference paper from the ontology *kmi-impact-ontology* to the KMi semantic web portal ontology. Some slots' values are directly mapped from the slots of the source class. Others are generated from further mappings.

The mapping *im2* is invoked when generating values for the slot *has-publication-reference* in mapping *im1*. On the one hand, the generated instance of *im2* will be referenced as one value of the slot *has-publication-reference* of the target instance of the mapping *im1*. On the other hand, the target instance of the mapping *im1* serves as the value of the correspondent slot specified in *im2* which is the slot *refers-to-publication*. Thus, the semantic relations between these target instances are established.

The mapping *im3* is invoked when generating values for the slot *included-in-publication* in *im1*. This mapping is an *n:1* mapping. This is because a number of conference papers may be published in the same conference, and thus should be included in the same conference proceedings. Hence, the mapping *im3* produces a new instance for the class *Conference-proceedings* for the specific conference only when the instance does not exist. Otherwise, it adds the instance of the conference paper as one value of the slot *contains-publication*, which is the correspondent slot of this sub mapping.

Mappings *im4* and *im5* are invoked by the mapping *im3* when mapping the values for the slot *has-publication-reference* and the slot *refers-to-event*. Like the mapping *im3*, these mappings are also *n:1* mappings.

Now let us investigate two conference papers *kmi-impact-item-552* and *kmi-impact-item-897* stored in the source data set. Both of them are published in the second International Conference of Knowledge Capture (KCAP-2003). Hence, there should be only one instance generated for each of the conference related classes, including *conference-proceedings*, *conference-proceedings-reference*, and *conference*. Figure 5 shows the mapping result of these two papers.

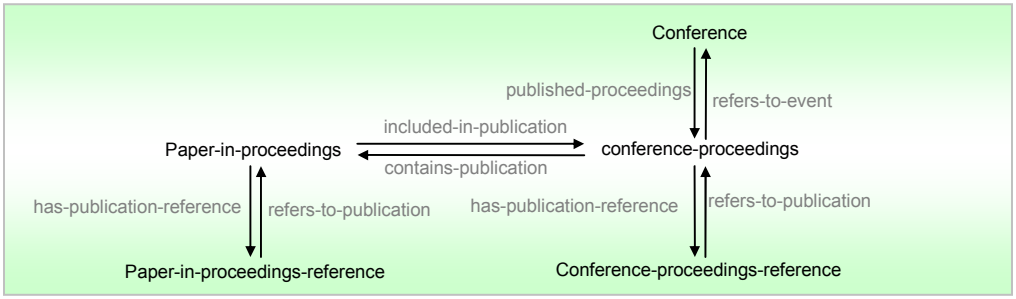

Figure 3. The relations of the class *paper-in-proceedings* with other classes in the target ontology from a conference paper's point of view

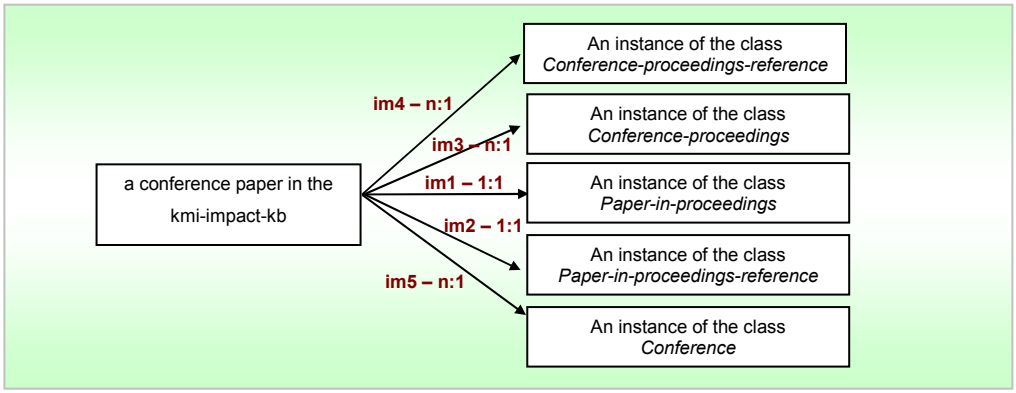

Figure 4. Mappings involved when mapping a conference paper in the source ontology to the target ontology

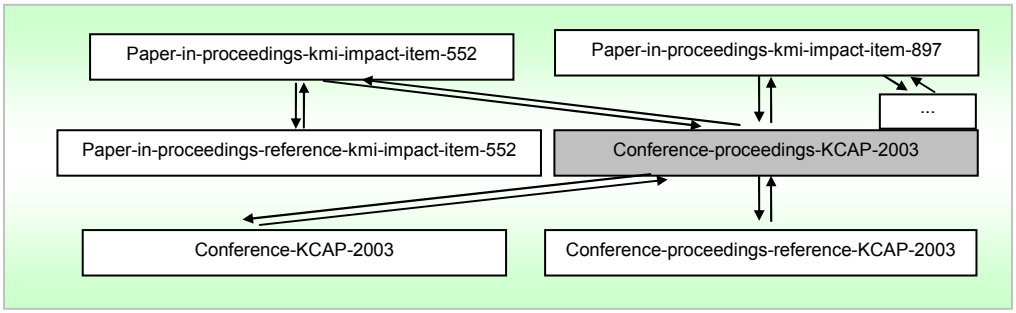

Figure 5. The mapping result of two papers which are both published in the conference KCAP-2003

Table 2. The major mappings involved in transforming a conference paper from the ontology *kmi-impact-ontology* to the KMi semantic web portal ontology

Instance mapping	Target class	Source Class	Constraints on source instances[6]	Target slot	Values from	Correspondent slot (only applicable for sub mappings)
im1	Paper-in-proceedings	Publications	The value of the slot *conference-name* should not be empty.	has-title	The source slot *has-title*	n/a
				has-first-author	The source slot *first-author*	
				has-other-authors	The source slot *other-authors*	
				has-publication-reference	The sub mapping *im2*	
				Included-in-publications	The sub mapping *im3*	
im2	Paper-in-proceedings-reference	Publications	n/a	has-title	The source slot *has-title*	refers-to-publication
				has-first-author	The source slot *first-author*	
				has-other-authors	The source slot *other-authors*	
				has-place-of-publication	The source slot *geographical-location*	
im3	Conference-proceedings	Publications	n/a	has-publication-reference	The sub mapping *im4*	Contains-publication
				refers-to-event	The sub mapping *im5*	
im4	Conference-proceedings-reference	Publications	n/a	has-place-of-publication	The source slot *geographical-location*	refers-to-publication
im5	Conference	Publications	n/a	has-pretty-name	The source slot *conference-name*	

[6] The constraints on sub mappings are implicit, as they work on those instances which invoke them.

RELATED WORK

As mentioned earlier, several approaches have been developed, which provide languages for specifying mappings between ontologies. Examples include Piazza [4], Madhavan et al. (2002) [6], and MAFRA. Among them, the MAFRA semantic bridging ontology is the closest one to our instance mapping ontology. As illustrated earlier, the MAFRA semantic bridging ontology is not as comprehensive as our ontology, as one important mapping relation is not captured, which allows the generation of slot values from instances. As a consequence, the transformed semantic data objects of the MAFRA mapping ontology contain few semantic relations. This makes the results less useful. Another difference between the MAFRA mapping ontology and our instance mapping ontology is that the MAFRA mapping ontology does not support the specification of the associated data sets. Thus, data transformation (or integration) engines will have difficulty accessing them without the help of further specification.

Piazza provides a language for mediating between data sources represented in XML and RDF [14]. Unlike the Piazza language, our instance mapping ontology does not limit itself to any particular representations. Another difference between the Piazza mediating language and the instance mapping ontology is that the Piazza mediating language is a query-based language which relies on queries to specify how to map data objects from source representations. Thus the specification is not declarative, and thus very hard to re-use and maintain. Finally, as the Piazza mediating language focuses on mediation, it does not address the generation of rich semantic relations between instances.

Madhavan et al (2002) proposed a framework for defining languages for specifying mappings. It is very different from our instance mapping ontology, as the framework focuses on the mapping of classes and concepts while ours concentrates on the mapping of instances.

The approach proposed by Stojanovic et al (2002) [11] addresses the transformation of data objects from relational database schemas into ontologies. It relies on a set of rules to achieve its task. However, no explicit vocabularies are provided to support the representation of mappings. The generation of semantic relations between the transformed ontologies has also not been addressed.

Finally, the semantic standard language OWL provides constructs to allow the specification of ontology mapping, e.g., *equivalentClass*, *equivalentProperty*, *sameAs*, *differentFrom*, and *AllDifferent*. These constructs are only able to specify simple mappings between concepts and instances. It can not be used to specify mappings that allow the transformation of data from one ontology to another.

CONCLUSIONS

In this paper we have presented an instance mapping ontology which provides comprehensive support for the transformation of data objects from source representations into ontologies.

Firstly, the instance mapping ontology allows the specification of mappings to be represented in a declarative and reusable format. Thus, robust tools can be constructed which on the one hand assist users to generate, discover, and maintain mappings at design time and on the other hand perform semantic data transformation at run time.

Secondly, the instance mapping ontology captures the complex mapping relations that are required to transform data from different representations into the target ontology. It distinguishes three ways to generate values for slots of the target instances, including i) from the values of source slots, ii) from existing instances, and iii) from the instances that are to be created in further mappings. In particular, the latter two ways support the generation of rich semantic relations between the target instances.

Finally, the instance mapping ontology is representation independent. This is supported by the construct *DataSet* which allows the specification of how to access the source data sets and the target data set. Thus, a data transformation engine is able to access data sets with the support of functionalities provided by the specified database or knowledgebase management system. However, in the cases when there is no management system available for accessing the specified data set, the data transformation engine will have to provide such services.

As discussed earlier, the instance mapping ontology has been applied in the KMi web site, transforming the underlying heterogeneous data sources into the specified domain ontology. The transformed semantic data can be accessed at http://semanticweb.kmi.open.ac.uk.

ACKNOWLEDGEMENTS

This research was partially supported by the Advanced Knowledge Technologies (AKT) project. AKT is an Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

REFERENCES

[1] Berners-Lee, T., Hendler, J., and Lassila, O., "The Semantic Web", Scientific American, May (2001).

[2] Doan, A., Madhavan, J., Domingos, P. and Halvey, A., "Learning to map between ontologies on the semantic web", in Proceedings of WWW'02, pp. 662-673 (2002).

[3] Domingue, J., "Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web", in Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada (1998).

[4] Halevy, A., Ives, Z., Mork, P., and Tatarinov, I., "Piazza: Mediation and integration infrastructure for semantic web data", Journal of Web Semantics, Vol. 1 (2), pp. 155-175, February (2004).

[5] Kalfoglou, Y. and Schorlemmer, M., "Ontology mapping: the state of the art", The Knowledge Engineering Review, Vol. 18 (1), pp. 1-31 (2003).

[6] Madhavan, J., Bernstein, P. A., Domingos, P., and Halevy, A., "Representing and reasoning about mappings between domain models", in Proceedings of the 18th National Conference on Artificial Intelligence (AAAI'02), Edmonton, Alberta, Canada, pp. 80-86 (2002).

[7] Maedche, A., Motik, B., Silva, N. and Volz R., "MAFRA - a mapping framework for distributed ontologies", in Proceedings of the13th International Conference on Knowledge Engineering and Knowledge Management, Sigüenza, Spain (2002).

[8] Motta, E., "Reusable Components of Knowledge Modelling: Case Studies in Parametric Design Problem Solving", IOS Press, Amsterdam (1999).

[9] Noy, N. F., "Semantic Integration: A Survey of Ontology-Based Approaches", SIGMOD Record, Special Issue on Semantic Integration (2004).

[10] Noy, N. F. and Musen, M. A., "PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment", in Proceedings of the National Conference on Artificial Intelligence (AAAI) (2000).

[11] Stojanovic, L., Stojanovic, N., and Volz, R., "Migrating data-intensive web sites into the semantic web", in Proceedings of the 17th ACM symposium on applied computing (SAC), ACM Press, 2002, pp. 1100-1107.

[12] Zhu, J., Uren, V., and Motta, E., "ESpotter: Adaptive Named Entity Recognition for Web Browsing", in Proceedings of Workshop on IT Tools for Knowledge Management Systems at WM2005 Conference, Kaiserslautern, Germany (2005).

[13] W3C, "OWL web ontology language guide", available online at: http://www.w3.org/TR/2003/PR-owl-guide-20031215/#OntologyMapping (Accessed 5 May 2005).

[14] W3C, Resource Description Framework (RDF) Model and Syntax, W3C Proposed Recommendation, available online at http://www.w3.org/TR/PR-rdf-syntax/ (Accessed 5 May 2005).

[15] W3C, Extensive Markup Lauguage (XML), available online at http://www.w3.org/XML/ (Accessed 5 May 2005).