Mapping Technology for Enterprise Integration

Borys Omelayenko **Dieter Fensel** **Christoph Bussler**

Vrije Universiteit, De Boelelaan 1081, 1081HV, Amsterdam, The Netherlands
Email: borys@cs.vu.nl, dieter@cs.vu.nl
Homepage: www.cs.vu.nl/~borys, www.cs.vu.nl/~dieter

Oracle Corporation, Redwood Shores, CA 94065, USA
Email: chris.bussler@oracle.com

Abstract

The emerging Semantic Web may change business integration from using partial, ad-hoc and costly connections to a qualitatively new level providing universal representation and transformation means for business terminologies, documents, business processes, services, and business models. Currently, the Semantic Web is based on representation languages like XML, RDF (Schema), and OIL, however it lacks proper transformation initiatives. In the paper we present an RDFT mapping ontology and show it's applications to business integration tasks.

1. Introduction

Business integration may become a killer application for the Semantic Web technology. Existent ad-hoc integration solutions become not sufficient to serve Web-based business integration. A large and exponentially growing number of enterprises to be integrated on the Web and high complexity of the integration transactions require development knowledge-intensive and highly automated integration technologies.

Semantic Web provides several basic techniques to represent knowledge-intensive structures to be used for integration: RDF[1], RDF Schema[2], DAML+OIL[3]. RDF is a language for representing data models in a form of object-property-value triples; it has several XML serializations that contain links to the underlying conceptual models. In turn, RDF Schema provides the basic type system for RDF data models by specifying classes of objects and their taxonomy, legal configurations and properties. However, it provides no formal semantics, no knowledge-intensive modeling primitives and no associated inference mechanism. Finally, the DAML+OIL ontology language provides an extension to RDF Schema (Broekstra et al, 2001) with well-defined semantics, expressive modeling primitives, and efficient reasoning support.

In addition to representation means, the business integration domain needs mature transformation languages and tools. The standard XML transformation language XSL-T (Clark, 1999) provides low-level transformation means operating at the level of XML elements and string values. Advanced business integration scenarios require transformation means dealing with high-level structural and ontological concepts.

However, at present time the Semantic Web does not provide them. The Triple language (Sintek & Decker, 2001) is proposed as a new RDF (Schema) query, inference and transformation language. Basically, the idea of Triple is in translating RDF structures into Horn logic. This allows performing Prolog-like rule creation and using any Horn-based inference engine. FaCT[4] is the standard classification-based reasoner for DAML+OIL, which can also be used for transformation. It is based on concept classification and requires the transformation task to be reformulated as a classification task. Classification and inference systems suffer severe performance decreases while operating with (hundreds) thousands of instances that can be a usual case for the business integration systems.

We propose an RDF mapping meta-ontology that contains several transformation primitives with limited expressiveness, that is sufficient to perform the integration tasks and still allows building efficient transformation engines.

We sketch the generic scheme for business integration in Section 2, introduce the RDFT meta-ontology in Section 3 and discuss its application to three important integration tasks: content integration (Section 4), document integration (Section 5), and process integration (Section 6). The paper ends up with final conclusions and future research directions.

2. The Scheme for Business Integration

The number of enterprises to be integrated with a single service can be quite high. For each pair of enterprises wiling to exchange some information the service needs be able to create necessary transformation rules and hence to maintain large and exponentially growing number of connections. This number is reduced to linearly growing by establishing a mediating data format maintained by the service together with mediating conceptual models for the data to be translated and ontological constraints on it.

[1] http://www.w3.org/TR/REC-rdf-syntax
[2] http://www.w3.org/TR/rdf-schema
[3] DAML+OIL soon will be replaced by the OWL, a W3C standard ontology language for the future Web.
[4] http://www.cs.man.ac.uk/~horrocks/FaCT/

For each pair of enterprises that need to translate their XML documents the service performs several transformation steps as depicted in Figure 1 and described with more details in (Omelayenko & Fensel, 2001). As shown in the figure the source XML document is first translated into its RDF data model, where two vocabularies: of XML element or attribute names and their values are separated. Accordingly, different XML element or attribute names are mapped to RDF Schema classes and properties. The documents are assigned to certain points in the source business process. These processes are aligned to the mediating process, as well as the documents and vocabularies are mapped to the mediating documents and vocabularies. All of them are represented in RDF Schema and thus at this step only RDF Schema classes and properties need to be mapped. Finally, the data model of the target document is constructed from the mediating representation and mapping RDF classes and properties of the target data model to the target XML syntax produces the specific target XML serialization.

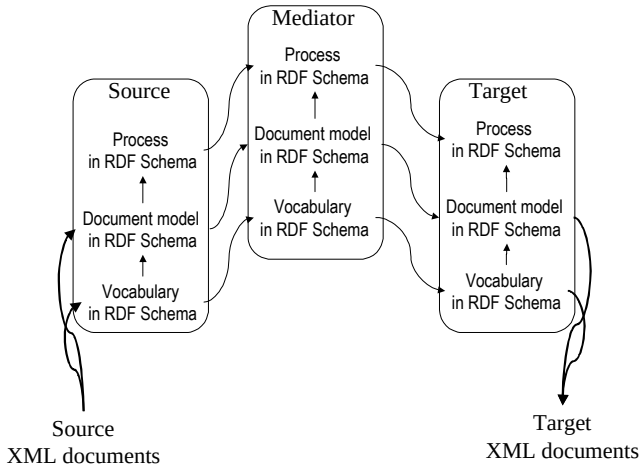

Figure 1: The business integration framework

3. RDFT Meta-Ontology

The RDFT (RDF Transformation) mapping meta-ontology[5] specifies a restricted language for mapping XML DTDs to/and RDF Schemas specially targeted for business integration tasks. The basic class diagram is presented in Figure 2, where the classes are represented by their names, and name nesting indicates the is-a relationship. The main concept of RDFT is the bridge between two sets of rdf:Resources (two sets of concepts), one of which is regarded as the source set, and the other one as the target set.

The bridges are grouped into maps. Each **Map** is a collection of bridges serving a single purpose. The maps are identified by their names (URL's) and they form minimal reusable modules of mapping information.

[5] http://www.cs.vu.nl/~borys/RDFT

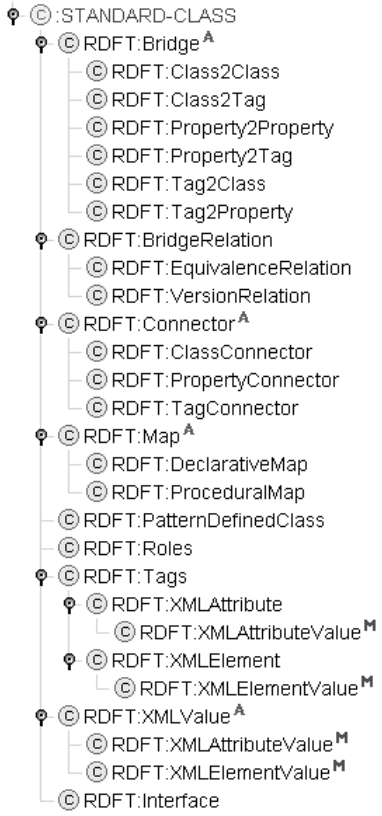

Figure 2: RDFT class diagram

An abstract class **Bridge** describes common properties of bridges allowing only one-to-many and many-to-one bridges. Each Bridge contains the ValueCorrespondance property linking a Map that specifies the necessary transformations of instance values of the source and the target concepts linked by the bridge.

The bridges also contain the Relation property linking to one of the BridgeRelations: EquivalenceRelation or VersionRelation:

- Equivalence bridges specify that the source element of a one-to-many bridge is equivalent to the target set of elements, and the source set of elements is equivalent to the target element for many-to-one bridges.
- A Version bridge specifies that the target set of elements form a (later) version of the source set of elements. Opposite to equivalence bridges, they assume that both source and target concepts belong to the same domain (or document standard), and may refer to two concepts with the same name (but different namespaces indicating versions), and imply that all the relations that hold for the original concept must hold for the versioned concept, if the opposite is not stated explicitly.

Several types of Bridges are defined in RDFT:

- Class2Class and Property2Property bridges between RDF Schema classes and properties, correspondingly. A one-to-many Class2Class bridges indicates that one instance of the source class is equivalent to the set of several instances, one instance of each target class.

Property2Property bridges have similar semantics applied to properties. In RDF Schema classes are represented by their names, place in taxonomy, and properties that are attached to this class. Properties are defined as first-class objects together with classes. All instance data of a document is represented with RDF properties and RDF classes just specify aggregation of properties. We did not included class-to-property and property-to-class bridges in RDFT to focus it at instance data transformation.

- Tag2Class and Tag2Property bridges between the source XML tags and the target RDF Schema classes and properties.
- Class2Tag and Property2Tag bridges between RDF Schema classes and properties, and XML attributes and elements of the target XML format.

All of the bridges contain the ValueCorrespondance property inherited from the abstract class Bridge linking to a Map. Two types of Maps are defined in RDFT:

- DeclarativeMap that specifies a set of bridges between discrete terms that may occur as the source and target instance values.
- ProceduralMap specifies an XPath (Clark, 1999) expression transforming instance data. A ProceduralMap is used when the data cannot be represented with sets of distinct terms, and thus, DeclarativeMaps cannot be constructed. XPath defines the means for two tasks: addressing data elements in XML documents and performing element or attribute value transformations (Chapter 4 of the specification[6]). In procedural maps we use only the second part of the XPath functions (e.g. substring_before).

The maps receive the input values and return the output results as formal parameters. To model this we need to distinguish between two properties with the same name but assigned to different classes (e.g. property Input of the class Map1 from the property Input of another map). It is not possible in RDF Schema where the properties are defined as first-class objects together with classes. For example, the property Code assigned to the class ProductDescription where it stands for a product code has a different meaning comparing to the property Code assigned to the class Country where it stands for a country code. We link formal parameters of the maps to the bridges with Connectors: ClassConnector, PropertyConnector, and TagConnector. The Connectors assign individual identifiers to each assignment of a property to a class.

Each Map has a certain Condition that is evaluated prior to execution of the bridge, and the bridge is executed only if the Condition is satisfied. For example, a one-to-many Property2Property bridge that splits a single string into two may have several maps attached: one is used if the two target strings are comma-separated in the source string, another is used if they are tab-delimited, etc.

Finally, RDF Schema provides a built-in way to model user's classes with rdfs:Class and properties with rdf:Property. To model XML elements and attributes we need to introduce the classes XMLElement and XMLAttribute. To model tag values that appear in XML DTDs we have introduced several XMLValues also presented in Figure 2.

We have developed RDFT accordingly to the Common Warehouse Model (CWM) Specification (CWM, 2001) by the Object Management Group[7] that provides a general architecture for a mapping ontology to be adopted by each specific mapping application. It contains primitives equivalent to Property2Property and Class2Class bridges, and their maps.

The expressive power of RDFT is intentionally kept quite limited to allow efficient compilation of RDFT maps into XSL-T[8] stylesheets allowing efficient transformation of instance data according to the bridges.

In the following sections we show the application of RDFT to different important business integration tasks: vocabulary integration, document translation, and process alignment.

4. Vocabulary Integration

We make our discussion on vocabulary integration mainly concerning product encoding vocabularies, or so-called content standards. Content standards provide hierarchies of product descriptions and define the subclass-of relationship between product categories. This allows modeling them as light-weight ontologies in RDF Schema (Omelayenko, 2001). Each product from a product catalog has an attached link to a certain product category, which actually describes the product and its properties.

The most well-known content standard UNSPSC[9] has a 5-level classification scheme with more than 17,000 categories. It is not descriptive, that is, it contains no attributes for products but only the hierarchy of product names. Another horizontal standard, Eclass[10], provides more than 13,000 categories with attributes tailored to the needs of industrial customers and their suppliers. One of the differences between them is that UNSPSC classifies the products from suppliers' view (when pen and paper belong to completely different groups) while Eclass imposes buyers' view (when both pen and papers belong to the same group).

Several subtasks arise in dealing with these standards on the Web.

Representing the standards on the Web in a standardized and universally accessible way may be done with RDF Schema. We represent the categories with RDF classes, the hierarchy of the categories with the rdfs:subClassOf relation, and category descriptions with rdfs:comment:

[6] http://www.w3.org/TR/xpath

[7] http://www.omg.org/

[8] www.w3.org/TR/xslt

[9] http://eccma.org/unspsc/

[10] www.eclass.de

```
<rdfs:Class rdf:about="eclass:24-01-04" rdfs:comment="Printers,
Plotters, Scanners">
  <rdfs:subClassOf rdf:resource="eclass:24-01"/>
</rdfs:Class>
```

Representing maps between the standards includes representing one-to-many and many-to-one equivalence between product categories. One-to-many RDFT bridges are used to represent equivalence a single category of the source standard to a set of target categories. For example, the Eclass category 24-01-04 Printer, Plotters, Scanners corresponds to three UNSPSC categories 43-17-22 Scanners, 43-17-25 Printers, and 43-17-25-01 Plotters (as depicted in Figure 3). This can be represented in RDFT with the following class-to-class bridge:

```
<rdft:Class2Class rdf:about="UNSPSC-Eclass Printers"
RDFT:Relation="EquivalentToSet">
  <rdft:SourceClass rdf:resource="eclass:24-01-04"/>
  <rdft:TargetClass rdf:resource="unspsc:43-17-22"/>
  <rdft:TargetClass rdf:resource="unspsc:43-17-25"/>
  <rdft:TargetClass rdf:resource="unspsc:43-17-25-01"/>
  <rdfs:subClassOf rdf:resource="rdfs:Resource"/>
</rdft:ClassBridge>
```

The bridge links the classes form two ontologies, Eclass and UNSPSC, marked with correspondent namespaces (their syntax is slightly modified for representational needs).

Finally, RDFT bridges being applied to the content integration task denote equivalence between sets of categories, and their execution results at reclassification of the product descriptions.

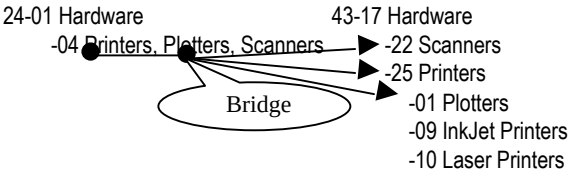

Figure 3: Two content standards

Let us continue with the documents that use these vocabularies.

5. Document Integration

Different formats for business documents (e.g. purchase orders, purchase order requests, etc.) are used by different applications inside a company and by company partners. A number of standards specify recommended XML format for the documents, e.g. cXML[11] and xCBL[12]. Even conceptually equivalent documents may have different XML serialisation, which creates the document transformation problem. Creation XSL-T rules linking different formats seems to be an evident solution for this problem. However, the data model of string tag values organized into XML trees adopted by XSL-T is not suitable for complicated document integration. As a result, an attempt to perform direct document integration with XSL-T fails because several complicated tasks might be resolved in a single set of rules:

- Transformation between different XML serializations.
- Matching different terminologies of XML tag names.
- Aligning different document data models for documents.
- Converting different terminologies of document tag values.
- Aligning various decompositions of information between documents and document roles in the underlying business processes. It is nearly impossible to solve this task with XSL-T.

Concisely, RDFT provides a higher-level add-on to XSL-T with additional support of separate mapping XML serializations, terminologies of tag names, vocabularies of tag values and different document data models.

Shifting from the data model of XML trees to document conceptual models and separated vocabularies is performed at three steps, as presented in Figure 1. First the source documents are transformed from XML representation into their data models in RDF by aligning document's DTDs to their RDF Schemas. Second, document integration is performed at the level of RDF Schemas by the means of RDFT. Finally, the resulting RDF Schemas are then translated to the target XML serialization.

Representing the maps between different documents can be easily done with RDFT at the level of conceptual models. For example, cXML allocates a single tag to represent street name and house number in a (delivery) address, while xCBL allocates two separate tags for street name and for house number. Such correspondence can be represented in RDFT with the following one-to-many Property2Property bridge:

```
<RDFT:Property2Property rdf:about="POstreetaddress"
   RDFT:Relation="EquivalentToSet">
  <RDFT:TargetProperty rdf:resource="ShipToHouseNumber"/>
  <RDFT:TargetProperty rdf:resource="ShipToStreet"/>
  <RDFT:SourceProperty
rdf:resource="ShipToPostalAddressStreet"/>
  <RDFT:SplitMergePattern rdf:resource="RE001"/>
  <RDFT:RangeBridge rdf:resource="rdft:EquivalenceOfLiterals"/>
  <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</RDFT:Property2Property>

<RDFT:RegularExpression rdf:about="RE001"
   RDFT:Expression="substring_before($source,',')">
  <rdfs:subClassOf rdf:resource="rdfs:Resource"/>
</RDFT:RegularExpression>
```

It is easy to see that RDFT separates mapping the terminology of document elements by explicitly representing it with the SourceProperty and TargetProperty from the map between the terms of instance values. Non-declarative transformations that are not supported by

[11] www.cXML.org
[12] www.xCBL.org

RDFT are represented with XPath value transformation expressions. However, declarative transformations supported by RDFT dominate in the business integration tasks.

Execution of the bridges relies on the following information: (i) SplitMergePatterns that explicitly specify the transformations to be performed; (ii) VocabularyMaps that specify how the (regular) property vocabularies must be converted.

Document appearance and consequent transformations are guided by document's position in the underlying business processes.

6. Process Integration

The process integration task is traditionally discussed as the task of mapping private business processes to a shared public process.

Public processes are visible to the trading partners, who had agreed on them and strictly follow in order to exchange messages over networks like the Internet. The processes as well as the message formats and contents are formally defined by B2B protocols like RosettaNet[13] or ebXML[14]. Private processes are invisible for trading partners and are maintained by a back-end application system like an enterprise resource planning system (ERP). The public processes and the private processes are both executed by a process integration system and have to be integrated with each other (Bussler, 2002).

Public processes as well as the message format definitions are commonly expressed in XML. Like in the document integration case, it results into a transformation problem that is difficult to manage based on XML. Therefore, the representation of processes and message formats in RDF is preferable in order to enable the transformation task.

A 'public' message received from a trading partner has to be processed and inserted into the private process, and a 'private' response might be extracted from the ERP and sent back. A good example is the exchange of purchase order (PO) requests and subsequent PO acknowledgement (POA) replies.

Two main transformation problems have to be addressed in a process integration system:

- Messages received in a format defined by a B2B protocol have to be transformed into back end application system format (transformation of PO into Order Request in Figure 5).
- The back end application system of a business may not provide a response message. It must be simulated by the process integration system (see Figure 5: a POA needs to be generated for an incoming PO).

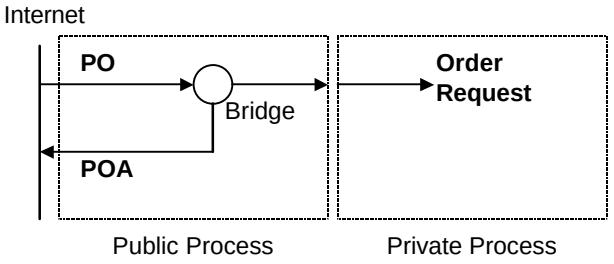

Figure 5: Process integration

An attempt to specify process transformation rules directly leads to the following problems: instead of separate definition of public and private processes, the overall processes need to be defined. This leads to an explosion since all possible combinations of internal and external processes have to be explicitly represented in single processes.

To perform efficient transformation we need to represent the processes with their input and output ports (represented with ports A-E in Figure 4). Each of the ports has the following structure:

- Port Name that is a unique identifier of the port.
- Message Type that specifies the kind of the message transformed via the port (e.g. notification, approval, etc.)
- Document Type that specifies a specific document type passed with the message (e.g. Purchase Order).
- TimePoint of the message that relates the message to a certain time point in the underlying process ontology

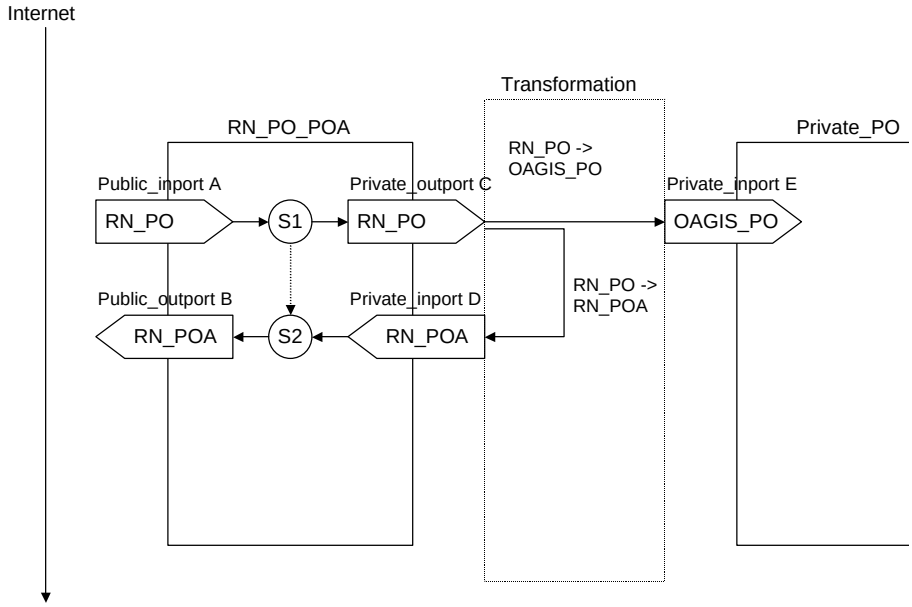

Figure 4: Modeling process integration: public and private processes aligned via ports

[13] www.rosettanet.org
[14] www.ebxml.org

(e.g. based on PSL[15]) and specifying message precessors and successors.

The specification of the process integration scenario consists of these ports aligned via TimePoints. To align two processes we need to specify ordinary RDFT bridges between the names, message and document types. Each RDFT map has a certain Condition that is evaluated before the map is going to be executed, and these conditions invoke necessary bridges that generate the messages to be simulated by a process integration system.

7. Conclusions

The expectations from the Web as a means for enterprise integration make traditional ad-hoc integration technologies insufficient and demand development new scalable and labour-saving techniques. For this the overall business integration task can be naturally decomposed into several subtasks: process integration, document translation, and aligning different complicated terminologies (e.g. content standards).

RDF (Schema) can be successfully used to represent the processes, documents, and terminologies, and a technology to link all of them together is needed. We developed RDFT, and RDF (Schema) mapping meta-ontology that allows representing the bridges of all these three different types in a unified way. We intentionally made the expressive power of RDFT quite limited to allow efficient compilation of RDFT bridges into XSL-T that performs actual transformation of instance data. We illustrated sufficiency of the expressive power with the examples showing how RDFT can be applied to solve typical vocabulary, document, and process transformation tasks.

A couple of future research question remains open:

- Searching for the bridges in content standards becomes a challenge because the standards have huge size and do not provide formal description of the categories.
- Creation a minimal and consistent set of bridges becomes the second challenge. Inference-based techniques may be used to check the consistency and redundancy of a particular RDFT map. The maps are represented as ontologies and this allows direct application of many inference engines.
- The semantics of RDFT classes is specified in terms of class and property instances, and expressing this semantics in formal ontology languages, like DAML+OIL remains and open task.

Finally, extensive case studies and successful application experience will judge about the overall utility of the approach.

Acknowledgements. We would like to thank Michel Klein, Heiner Stuckenschmidt and Volodymyr Zykov for their helpful discussions.

[15] http://www.mel.nist.gov/psl/

References

Broekstra, J., Klein, M., Fensel, D., Decker, S., van Harmelen, F., and Horroks, I.; 2001. Enabling knowledge representation on the Web by extending RDF Schema, In *Proceedings of the 10th World Wide Web Conference*, Hong Kong, China, May 1-5.

Bussler, C.; 2002. Modeling and Executing Semantic B2B Integration, In *Proceedings of the 12th International Workshop on Research Issues on Data Engineering: Engineering E-Commerce / E-Business Systems (RIDE-2EC'2002) (In conjunction with ICDE-2002)*, IEEE CS Press, San Jose, USA, February 24-25.

Clark, J.; 1999. XSL Transformations (XSL-T), *W3C Recommendation.* [http://www.w3.org/TR/xslt/]

CWM; 2001. Common Warehouse Model Specification, *Object Management Group*. [http://www.omg.org/cwm/]

Omelayenko, B.; 2001. Preliminary Ontology Modeling for B2B Content Integration, In *Proceedings of the First International Workshop on Electronic Business Hubs at the Twelfth International Conference on Database and Expert Systems Applications (DEXA-2001)*, IEEE CS Press, Munich, Germany, September 3, 7-13.

Omelayenko, B. and Fensel, D.; 2001. A Two-Layered Integration Approach for Product Information in B2B E-commerce, In Madria, K. and Pernul, G. (eds.), *Proceedings of the Second International Conference on Electronic Commerce and Web Technologies (EC WEB-2001)*, Springer-Verlag, LNCS 2115, Munich, Germany, September 4-6, 226-239.

Sintek, M. and Decker, S.; 2001. TRIPLE - An RDF Query, Inference, and Transformation Language, In *Proceedings of the Workshop on Deductive Databases and Knowledge Management (DDLP-2001)*, Tokio, Japan, October 20-22.