Ontologically enhanced RosettaNet B2B Integration

Paavo Kotinurmi[1], Armin Haller[2], Eyal Oren[3]

[1]Helsinki University of Technology
P.O.Box 9210, 02015 TKK, Finland
Paavo.Kotinurmi@tkk.fi

[2]Digital Enterprise Research Institute
National University of Ireland, Galway, Ireland
armin.haller@deri.org

[3] Vrije Universiteit Amsterdam
AI Department, 1081HV Amsterdam, Netherlands
eyal@cs.vu.nl

Abstract

RosettaNet is an industry-driven e-business process standard that defines common inter-company public processes and their associated business documents. RosettaNet is based on the Service-Oriented Architecture (SOA) paradigm and all business documents are expressed in DTD or XML Schema. Our "ontologically-enhanced RosettaNet" effort translates RosettaNet business documents into a Web ontology language, allowing business reasoning based on RosettaNet message exchanges. This chapter describes this extension to RosettaNet and shows how it can be used in business integrations for better interoperability. The usage of Web ontology languages in RosettaNet collaborations can help accommodate partner heterogeneity in the setup phase and can ease the back-end integration, enabling for example more competition in the purchasing processes. It provides also a building block for a semantic SOA with discovery, selection and composition capabilities.

Keywords

RosettaNet, B2B integration, RosettaNet ontology, Semantic Web Services.

1 Current Situation

Information and communication technologies are increasingly important in the daily operations of organisations. In the current networked business environment, most information systems need to interoperate with other internal and external information systems. Such interoperation is not easily achievable and therefore causes significant costs. For example, Brunnermeier & Martin (2002) studied interoperability in the U.S. automotive supply chain and estimated the cost of poor interoperability in product data exchange alone to be around one billion dollar per annum.

Standards such as RosettaNet or ebXML facilitate Business-to-Business (B2B) integration (Shim et al., 2000). These standards support electronic commerce over existing Internet standards and lead to cost and extensibility benefits. The aim of B2B standards is to facilitate integration with less implementation effort for each e-business partner organisation. Many B2B standards employ XML technologies and the Internet to standardise document exchange and ease the implementation effort of collaborations (Nurmilaakso & Kotinurmi, 2004; Shim et al., 2000). However, there are many competing B2B standards that are not mutually interoperable. So the choice for a particular B2B standard also forms a potential integration bottleneck.

Emerging Semantic Web technologies enable a business integration that is more adaptive to changes that might occur over the lifetime of B2B integrations (Fensel, 2003; Trastour,Preist, & Coleman, 2003). This chapter will describe existing B2B standards and focus on an extension of RosettaNet that uses

Semantic Web technologies. The usage of this extension for B2B integration and the added flexibility that is gained with it will be demonstrated in a practical integration scenario.

2 Existing standards for B2B integration

Many relevant standards have been introduced to alleviate B2B integration issues such as data heterogeneity and process heterogeneity. We describe XML and RosettaNet, and explain the remaining issues in practical B2B integrations even when using these standards.

2.1 XML and RosettaNet

XML (Extensible Markup Language) is a language for describing and exchanging data. Before the introduction of XML, business partners needed to accommodate various file formats, such as flat files or different EDI (Electronic Data Interchange) versions, and setup a parsing/management infrastructures for each format used by a partner. The introduction of XML lowered the integration barriers between organisations, as partners could reuse their XML infrastructure for all exchanged documents between all partners. The main two schema languages associated to the XML standard are DTD (Document Type Definition language) and XSD (XML Schema Definition language). These schema languages enable business partners to validate whether incoming and outgoing documents conform to a required structure.

The use of XML as such does not resolve interoperability issues in B2B integrations, since the exchange of XML documents does not mean that the documents are understood similarly. Therefore, standards are needed that guide how XML is used in B2B integrations. RosettaNet[1] is one such XML-based B2B standard; already in 2004, RosettaNet had over 3000 documented implementations (Damodaran, 2004). Other common B2B standards include OAGIS[2], ebXML[3] and UBL[4].

RosettaNet is an industry-driven consortium that aims to create, implement, and promote open B2B integration standards. The member organisations represent the information technology, electronic components, semiconductor manufacturing, telecommunications, and logistics industries. The most important components in RosettaNet are Partner Interface Processes (PIPs), dictionaries and the RosettaNet Implementation Framework (RNIF) which are described in the following paragraphs.

2.1.1 RosettaNet Partner Interface Processes (PIPs)

RosettaNet Partner Interface Processes (PIPs) define cross-organisational processes (choreographies), which define the message exchange patterns occurring in different business collaborations. Each PIP contains a specification document, a set of document schemas and message guidelines to help to interpret these schemas. RosettaNet processes are not directly executable, but are manually mapped by trading partners onto their internal systems (Aalst & Kumar, 2003). RosettaNet PIPs are organised into eight clusters, denoted by numbers, which are further subdivided into segments, denoted by letters. For example, cluster 3 deals with “Order Management”; it is divided into four segments including for example PIP 3A “Quote and Order Entry”.

The specification document in a PIP defines the message exchange sequence using Unified Modeling Language (UML) activity diagrams, sequence diagrams and textual descriptions; the specification document also describes, in English, the roles of the partners and the conditions to initiate a collaboration. The document schema and the accompanying message guidelines define the valid PIP business document structure and content. The message guidelines introduce additional constraints and explanations, such as the meaning of a modification date and the representation and interpretation of date values. Similarly, the meanings of business codes that can appear in a document are specified.

Historically, PIPs have been expressed using DTD-based schemas. New PIPs, developed after 2004, use XML Schemas for documents and the ebXML Business Process Specification Schema (Clark et al., 2001) for the process descriptions. Since XML Schema is more expressive than DTD, some of the constraints in the message, which have been described only textually before, can now be represented formally. However, still in early 2008, many PIPs are specified only using DTDs and thus only contain these additional constraints in the accompanying English text.

Common terms used in all PIPs are defined in the global RosettaNet dictionaries. In addition to dictionaries, global RosettaNet identifiers are used, such as Data Universal Numbering System (DUNS) codes to identify companies and the Global Trade Identification Number (GTIN) to identify product types.

2.1.2 RosettaNet Implementation Framework (RNIF)

The RosettaNet Implementation Framework (RNIF) specifies secure messaging over the Internet. It defines the RosettaNet business message that contains the business document specified by the schemas and the necessary headers and security features needed to process the messages. RNIF also defines how attachments are encoded in the RosettaNet business messages and uses e.g. MIME and S/MIME for packing and encryption.

RNIF defines exception-handling mechanisms and ensures that the delivery is non-repudiated, so neither the sender nor the receiver can later deny having sent or received the RosettaNet business message. Many vendors, such as BEA and Microsoft, support RNIF in their products. The currently widely used RNIF version 2.0 does not use Web Service standards such as SOAP or WSDL. However, the RosettaNet Multiple Messaging Services specification, published in 2006, defines how RosettaNet PIPs can be sent using ebXML Messaging Services, Web Services or AS/2 specification.

2.2 Problems in RosettaNet integrations

The standardised business processes described by RosettaNet simplify cross-organisational collaboration and B2B integration. The degree of standardisation offered still requires considerable manual effort during the setup of a B2B collaboration (Trastour, Bartolini, & Preist, 2003). This manual effort is necessary given the modelling freedom left by weakly defined document schemas. As a consequence interoperability challenges are only partly addressed by the introduction of standardised PIPs and B2B integrations still suffer from long set up times and high costs (Damodaran, 2005; Preist et al., 2005).

To detail the challenges in a practical B2B integration situation, we present a quoting and purchasing scenario based on RosettaNet PIPs “3A1 Request for Quote (RFQ)” and “3A4 Request Purchase Order (PO)”. Figure 1 shows the overall choreography including the message exchange of a Buyer (requester) and a Seller (provider) using BPMN[5] notation. The white coloured activity boxes denote parts of the internal computational steps, dotted boxes are placeholders for possibly many computation steps performed internally, whereas the dark coloured boxes represent the public behaviour according to PIP 3A1 and PIP 3A4 respectively. The process message boxes here include RosettaNet-specific RNIF behaviour related to validating and acknowledging incoming messages.

The scenario presented in Figure 1 contains interaction with only one Seller. When the Buyer wants to introduce competition by receiving multiple quotes from different sellers, the following interoperability challenges arise:

- **Business process alignment** The Buyer's internal processes have to be adjusted to the introduction of a new Seller. Although all processes should conform to the same common PIP, manual alignment is still necessary, since the PIP's business process definitions are informal and lack process details.
- **Expressive power** The schema languages (DTD and XML Schema) lack expressive power to capture all necessary constraints and do not make all document semantics explicit. This lack of expressive power in the current RosettaNet specification has been acknowledged by RosettaNet experts (Damodaran, 2004, 2005).
  For example, the RosettaNet business dictionary defines a finite set of enumerated values for certain elements in a PIP message but does not describe the relationship between these values. This leads to heterogeneity in the messages as for example EU sellers typically use Euro values for quotation and the metric system for units of measurement, whereas U.S.-based companies quote in Dollars and inch-pound units.
- **Repetition** Current schema languages have limited support for information reuse through reference, rules, and external imports. As a consequence, messages contain a high amount of repetition since all implicit information must be stated explicitly. These repetitions make the schemas long and complex to interpret.

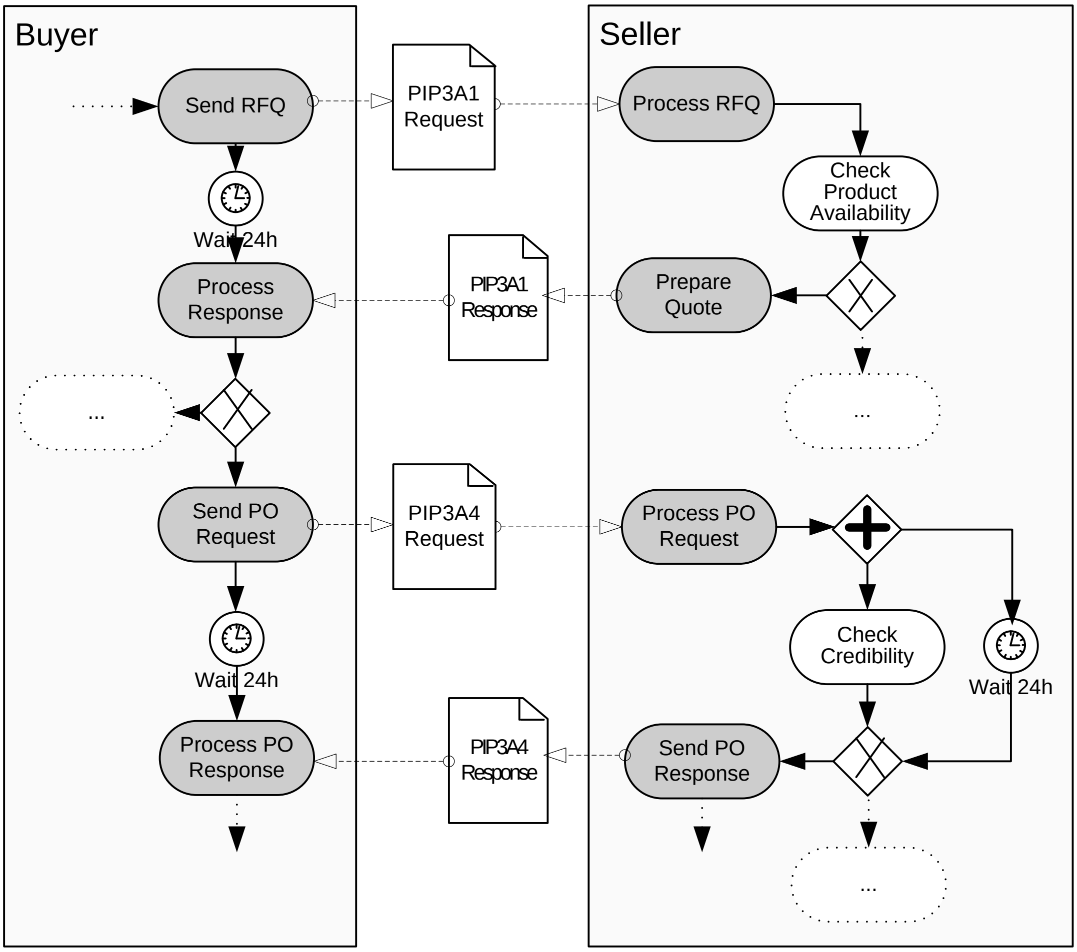

Fig. 1. RosettaNet B2B Collaboration in a quoting and purchasing process

- **Ambiguous message definitions** PIP message definitions are ambiguous: partners can use the same PIP messages in different ways. Logically similar information can be represented in many different places, for example on the level of "total order" or on "order-line item" level, and many elements within a schema definition for a PIP are optional and not implemented by every company.
  Thus, the Buyer might need to (manually) interpret additional information sent by a newly introduced Seller. For example, a new Seller might typically offer substitutable products in its responses which the Buyer can not handle without changes to the RosettaNet solution.
- **Context-dependent constraints** PIP processes expressed in XSD schemas are highly nested and the same elements may appear in several documents exchanged during an interaction. In fact, the nesting is an advantage over DTD-based PIPs which are self-contained and duplicate the same schema information in every PIP.
  But, as identified by Trastour, Preist, & Coleman (2003), constraints on a class in RosettaNet depend on its deployment context, including the PIP document being used, the trading partner and its role (buyer or seller) and the business process in which the messages are exchanged.
  In current B2B integrations, developers have to manually aggregate contextual constraints into refined XML schemas or into informal documents. Such manual integration is inefficient and leads to interoperability problems when partners apply different constraints

When the number of partners increases, these limitations become increasingly important, especially for companies that are not in a position to dictate their partners the usage terms of PIPs. Since resolving heterogeneities on a case-by-case basis is expensive, in practice the choice of integration partners is limited and fixed at design-time. As a consequence, B2B integrations typically do not support competitive arrangements in which multiple trading partners compete, at runtime, for best offers.

To overcome these challenges, we introduce semantic technologies to these B2B integrations. As such the information exchanged in the RosettaNet messages can be encoded in a formal Web ontology language, including definitional facts that make the dependencies between elements defined in the RosettaNet business dictionary explicit. A semantic Service-Oriented Architecture to communicate and mediate between the RosettaNet-enabled business partners and the internal back-end systems is applied as a proof-of-concept implementation model for the ontologically-enhanced RosettaNet approach.

3 The Ontologically-enhanced RosettaNet Solution

The objective of ontologising RosettaNet is to ease the integration to sellers and thus introduce more competition. The solution should be flexible to adding new partners and new collaborative processes and support coping with inevitable variations that come when working with multiple partners. A quick integration process, reuse of existing solutions and powerful discovery, selection, composition, validation and mediation capabilities are important objectives for the proposed solution. Companies have invested considerable amounts of money and resources to implement current B2B integrations based on existing B2B standards, and they have the supporting infrastructure largely in place. In our scenario, only the buyer adopts semantic technologies internally, enabling to manage integration with multiple sellers using current B2B standards. The Ontologically-enhanced RosettaNet solution shows how existing RosettaNet challenges can be solved using semantic technologies.

3.1 Semantic Technologies for B2B Integration

Semantic Web technologies and semantic Web Services have been proposed to achieve more dynamic partnerships (Bussler et al., 2002) and constitute one of the most promising research directions to improve the integration of applications within and across organisations.

Several standardisation efforts, such as OWL-S (Martin et al., 2004), WSMO (Roman et al., 2005) and SWSF (Battle et al., 2005), define frameworks and formal language stacks for semantic Web Services. Semantic Web Services employ ontologies for semantic annotation of B2B integration interfaces. Such semantic annotation enables (semi-)automated mediation of heterogeneous message content and more flexible service discovery, composition and selection (McIlraith et al., 2001).

Our solution demonstrates how a Service-Oriented Architecture (SOA) capable of handling semantic technologies can tackle some of the interoperability challenges with RosettaNet-based B2B integrations related to purchasing processes involving multiple partners.

We demonstrate our ontologically-enhanced RosettaNet solution using WSMO. The choice has been made on the fact that we require a more expressive language than RDF(s), but also a framework treating services as first-class citizen in its metamodel. However, although we have opted to use WSMO and its stack of WSML languages, the ontology could also be defined in OWL and OWL-S.

3.2 The WSMO Semantic Web Services framework

WSMO (Bruijn et al., 2005) is a Semantic Web Services framework consisting of a conceptual model and a language for describing the relevant aspects of services. The goal of such markup is to enable the automation of tasks (such as discovery, selection, composition, mediation, execution and monitoring) involved in both intra- and inter-enterprise integration. The WSMO framework is implemented in a SOA (Haselwanter et al., 2006; Vitvar et al., 2007) that can support RosettaNet integration scenarios (Haller et al., 2007).

The markup of services according to the WSMO conceptual model is expressed in the Web Service Modeling Language (WSML) family of ontology languages. WSML offers a human readable syntax, as well as XML and RDF syntaxes for exchanging data between services. WSML distinguishes conceptual and logical expression syntaxes. The conceptual syntax is used to model the fundamental WSMO concepts (Web Services, ontologies, goals and mediators); the logical expression syntax is used for describing additional constraints and axioms.

WSML consists of a number of variants based on different logical formalisms and correspond with different levels of logical expressiveness, which are both syntactically and semantically layered. These variants are WSML-Core, WSML-DL, WSML-Flight, WSML-Rule and WSML-Full. WSML-Core is the least expressive variant but with the best computational characteristics. WSML-Core is expressively equivalent to Description Logic Programs (Grosof et al., 2003), an intersection of Description Logic and Horn Logic. The WSML-DL, WSML-Flight and WSML-Rule variants extend WSML-Core to provide increasing expressiveness in the direction of Description Logics and Logic Programming. WSML-Full is the union of these two directions, making it the most expressive WSML variant.

3.3 RosettaNet Ontology

Ontologies, used for semantic annotation of the B2B integration interfaces, are formal descriptions of the concepts and relationships that can exist in some domain (Gruber, 1993; Uschold & Gruninger, 1996). The RosettaNet specification framework is such a world-view, namely that of the RosettaNet consortium, encompassing an exhaustive definition of the concepts, attributes and relations in the e-business domain.

However, the conceptualisation in form of DTDs and XML Schemas is not rigorously formal and leaves knowledge implicit in natural language. In this section, we show how to translate the PIPs into a formal Web ontology language. By translating the RosettaNet specifications to a richer and formal Web ontology language the constraints on the semantics of the business documents can be captured explicitly.

This section presents a generic RosettaNet ontology. As mentioned earlier, the ontology includes facts that cannot be expressed in the schema languages currently used in the RosettaNet PIPs. The ontology is modelled according to RosettaNet PIP specifications, in our example PIPs 3A1 and 3A4, containing concepts such as *PartnerDescription* or *PhysicalAddress,* and their attributes. Listing 1 shows a snippet of the RosettaNet ontology, describing how the *PartnerDescription* is expressed in the Web Service Modeling Language. The concept definition includes its properties and the cardinality constraints and type information for each property.

```
concept partnerDescription
    nonFunctionalProperties
        dc#description hasValue "The collection of business properties that describe a business identity."
    endNonFunctionalProperties
    globalbusinessidentifier ofType (1 1) _int
    globalsupplychaincode ofType (0 1) _string
    globalpartnerclassificationcode ofType (1 1) globalPartnerClassificationCode
```

Listing 1. Product ontology extract in WSML[6]

3.3.1 Definitional Facts

In this section we focus on the modelling of implicit knowledge in the RosettaNet specification, which can be made explicit in the ontology. For example, the RosettaNet business dictionary defines an enumerated list of 367 possible values for *units of measurements*, with the logical relationships between the values unspecified. They are directly modelled and constrained in RosettaNet as tokenised strings. All the inherent relations between the individual tokens are left unspecified. We made such logical relations explicit and included these axiomatisations in our ontology.

First, we identified for each tokenised string in the XML Schema its unit type class membership in the Suggested Upper Merged Ontology (SUMO) (Niles & Pease, 2001). SUMO is a richly axiomatised formal ontology created by the merger of multiple existing upper-level ontologies. SUMO is divided into eleven sections whose interdependencies are carefully documented. We are mostly interested in classes from the base ontology, numeric and measurement layer. Other parts of the ontology include among others, temporal, process and class theories. All unit types in the 367 token values can be related to the *PhysicalQuantity* class in SUMO, which itself subclasses *ConstantQuantity*, *FunctionQuantity* and *UnitOfMeasure*. By using SUMO we derive foundational relations, such as the equivalence of 1 *litre* to 1 *cubic decimetre* and the relation that 4.54609 *litres* are equal to 1 *UnitedKingdomGallon*, all facts defined in the axiom schemata of SUMO. Listing 2 shows the concept definitions in the ontology and its related classes in SUMO. *PoundMass* and *Kilogram* in SUMO are second level subclasses of *ProductQuantity*.

```
ProductQuantity subConceptOf sumo#ProductQuantity concept PoundMass
subConceptOf sumo#PoundMass concept Kilogram subConceptOf
sumo#Kilogram
```

Listing 2. *UnitOfMeasureTypes* in the RosettaNet ontology

Next, we identified common types in the tokens and modelled them as concepts in the ontology. Examples of similar tokens are a *10 Kilogram Drum*, a *100 Pound Drum* and a *15 Kilogram Drum*. Listing 3 shows the first two tokens in its XML Schema definition and listing 4 shows its representation in our ontology. We identified the concept *Drum* as being member of a *FluidContainer* in SUMO and it inherits similarly to all other converted unit types the *hasTokenValue*, *hasUnitSize* and *hasUnitType* attributes from its parent concept (*Quantity*).

```
<xs:simpleType name="UnitOfMeasureContentType">
 <xs:restriction base="xs:token">
  <xs:enumeration value="1KD">
   <xs:annotation>
    <xs:appinfo>
     <urss:Definition>10 Kilogram Drum.</urss:Definition>
    </xs:appinfo>
   </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="1PD">
   <xs:annotation>
    <xs:appinfo>
     <urss:Definition>100 Pound Drum.</urss:Definition>
    </xs:appinfo>
   </xs:annotation>
  </xs:enumeration>
 </xs:restriction>
</xs:simpleType>
```

Listing 3. UnitOfMeasureTypes extract of XML Schema

```
concept ProductQuantity subConceptOf sumo#
    ProductQuantity
  hasTokenValue ofType _string
  hasUnitQuota ofType _float
  hasUnitType ofType ProductQuantity

concept Drum subConceptOf {sumo#FluidContainer,
    ProductQuantity}

instance _10KilogramDrum memberOf Drum
  hasTokenValue hasValue "1KD"
  hasUnitQuota hasValue 10
  hasUnitType hasValue Kilogram

instance _100PoundDrum memberOf Drum
  hasTokenValue hasValue "1PD"
  hasUnitQuota hasValue 100
  hasUnitType hasValue PoundMass
```

Listing 4 UnitOfMeasureTypes extract in the RosettaNet ontology

This style of modelling allows us to further include semantic relations between instances of the same unit type concept. To define the numerical dependencies between different *UnitOfMeasureContentType* we add equivalence relations similar to the one shown in listing 5. It states that a *Quantity* instance with a certain amount ?z of "100 Pound Drum" unit types equals 4.5359237 times *10 Kilogram Drums*. Since we have derived a subsumption hierarchy, this axiom applies to all sub-classes of *Quantities*, such as *ProductQuantity*, the fourth most used type in the RosettaNet schema.

```
axiom _1PD1KDDependency
  definedBy
    ?x memberOf Quantity and ?x[hasUnitType hasValue _100PoundDrum] and ?y[hasNumericalValue hasValue ?z]
    equivalent
    ?x memberOf Quantity and ?x[hasUnitType hasValue _10KilogramDrum] and ?y[hasNumericalValue hasValue
        wsml#numericMultiply(?z1,?z,4.5359237)].
```

Listing 5. Equivalence relation between 100 Pound Drum and 10 Kilogram Drum

Since RosettaNet does not and can not model all domain knowledge, such as the actual product and partner identification, measurement types or currency types, other specifications (ontologies) have to ensure the uniform understanding of the elements within a message. RosettaNet publishes guidelines which standards to use, but does not reference or integrate ontologies to homogenise their usage.

ProductIdentification and *PartnerIdentification*, two element types highly used throughout the RosettaNet specifications (see Haller et al. (2008) for detailed statistics on the occurrence of schema types within RosettaNet) reference for example one of the following identifier types: *uat:identifiertype*, *ulc:AlternativeIdentifier*, *udt:GTIN*, *udt:DUNS*, *udt:DUNSPlus4* and *udt:GLN*. The role of these identifiers is to describe products (GTIN), companies (DUNS) and locations (GLN or DUNSPlus4 for company location) uniquely. When, for example, ordering a hammer, it is easier to refer to an identifier, such as the 14-digit GTIN “55566677788899”, than specifying “Tool, hammer, handtool, Fiskars,...” in every message. Alternative identifiers can also be used, typically for buyer identification. Using differing identification schemas creates a mapping challenge where ontologies can help to state similarity between different identifiers for a product.

Since an n-to-n mapping between identifiers is unfeasible even in an ontology, we propose to map to an existing product classification such as the eCl@ss classification (code \AAA374002" for hammer). By specifying this semantic detail and referring to the existing eClassOwl-ontology[7] (Hepp, 2005b), it is possible to provide information on similar products, such as hammers produced by other manufacturers. Further benefits materialise when respective UN/SPSC classifications are mapped to eCl@ss categories. This enables companies that map their identifier codes to a category to provide substitutable products and increase their chance of realising sales.

3.3.2 Domain-specific Rules

Each collaboration requires the setup of additional domain-specific rules to capture any data heterogeneity that is not resolved by the definitional facts in the domain ontology.

These domain specific rules define how attribute values in the different ontology instances are related. One such example is given in listing 6. It defines a constraint how a unit price relates to the *financialAmount* and *productQuantity* properties and how from a given instance a per-unit price can be calculated. The constraint can be used by the requester to implement a price comparison between competing offers received from different partners. The dependencies between the different packaging sizes and its corresponding values are made explicit in the ontology. This simple illustrative example shows how two partners can agree on the interpretation of the *financialAmount* in a PIP 3A1 message instance, since the amount can refer to different quantities of the product. The RosettaNet specification includes many more elements where the relation between values is not as straightforward as in the unit price example such as the payment terms, shipment terms or tax types to just mention some of the under-specified elements in a RosettaNet PIP 3A1.

```
relation unitPrice (ofType financialAmount, ofType productQuantity, ofType _decimal)
  nfp
    dc#relation hasValue unitPriceDependency
  endnfp

axiom unitPriceDependency
  definedBy
    !- unitPrice(?x,?y,?z) and wsml#numericDivide(?z,?x,?y).
```

Listing 6. Unit price constraint

3.4 Language Expressivity Advantages in Ontologised RosettaNet

This section presents our transformation methodology to translate traditional XML Schema and DTD-based RosettaNet messages to our ontological schema; this methodology can be used to non-obtrusively introduce the solution to existing B2B collaborations.

The DTD versions of PIP 3A1 and PIP 3A4 support two different kinds of product identifiers; the Global Trade Identification Number (GTIN), which is recommended by RosettaNet, and company-specific identifiers. The extract in listing 7 shows the definition of product identifiers in the PIP 3A1 (and 3A4). The PIP3A1 DTD is very long so only the relevant lines (291-304) are shown.

```
<!ELEMENT ProductIdentification
  (GlobalProductIdentifier?,
   PartnerProductIdentification*)>

<!ELEMENT GlobalProductIdentifier
  (#PCDATA)>

<!ELEMENT PartnerProductIdentification
  (GlobalPartnerClassificationCode,
   ProprietaryProductIdentifier,
   revisionIdentifier?)>

<!ELEMENT ProprietaryProductIdentifier
  (#PCDATA)>
```

Listing 7. PIP 3A1 DTD extract

RosettaNet message guidelines for PIP 3A1 add a *natural language constraint* for *ProductIdentification* that the DTD's expressive power does not capture: *Constraint: One instance of either GlobalProductIdentifier or PartnerProductIdentification is mandatory*. Without this constraint, a valid *ProductIdentification* could be without any identifiers as both identifications are optional. PIP 3A1 is not yet available as XML Schema, where such constraint can be expressed.

Listing 8 shows such an XML Schema version of PIP 3A4. The namespaces and annotations are dropped for brevity as XML Schemas take more space than DTDs. The XML Schemas name PIP elements differently to DTDs. The XML Schema also allows arbitrary authorities to specify the identification schemes, which introduces another mapping challenge.

```
<xs:element name="ProductIdentification" type="ProductIdentificationType" />
<xs:complexType name="ProductIdentificationType">
 <xs:complexContent><xs:sequence>
   <xs:element name="ProductName" type="xs:string" minOccurs="0" />
   <xs:element name="Revision" type="xs:string" minOccurs="0" />
   <xs:choice>
     <xs:element ref="AlternativeIdentifier" maxOccurs="unbounded" />
     <xs:element ref="GTIN" />
   </xs:choice>
 </xs:sequence></xs:complexContent>
</xs:complexType>
<xs:element name="AlternativeIdentifier" type="AlternativeIdentifierType" />
<xs:complexType name="AlternativeIdentifierType">
 <xs:sequence>
   <xs:element name="Authority" type="xs:string" />
   <xs:element name="Identifier" type="xs:string" />
 </xs:sequence>
</xs:complexType>
```

Listing 8. PIP 3A4 XML Schema extract

In the Web Service Modeling Language such constraints can be easily expressed as shown in this section. The product identifier information in our ontology is presented in listing 9. The GTIN is handled as any other identification authority/qualifier (*qualificationAgency*) and the RosettaNet DTD and XML Schema product identification information can be presented in this ontology including the natural language constraints. The qualification agency can be for example the *buyer's, seller's* or *original equipment manufacturer's identifier* or any other identification scheme provider. The constraint in Listing 9 ensures that the value of *qualificationAgency* is among those supported by the Buyer. The benefit of applying a more expressive Web ontology language such as WSML is that it allows the description of logical relationships between the elements. This information can subsequently be applied for better validation of the message contents. Furthermore, with ontologies it is straightforward to specify that a given product belongs to certain classification class and this information can be utilised to offer suitable substitutable products. There is existing work on ontologising product classifications (Hepp et al., 2005) available for reuse in this.

```
concept productIdentification
      nonFunctionalProperties
            dc#description hasValue "Collection of business properties describing identifiers."
      endNonFunctionalProperties
      productIdentifier ofType (1 1) _string
      qualificationAgency ofType (1 1) _string
      revision ofType (0 1) _string

axiom qualificationAgencyConstraint
      nonFunctionalProperties
            dc#description hasValue "The valid list of agencies who have defined product identifiers."
      endNonFunctionalProperties
      definedBy !− ?x[qualificationAgency hasValue ?type] memberOf productIdentification
            and (?type = "GTIN" or ?type = "Manufacturer" or ?type = "Buyer"
                 or ?type = "Seller" or ?type = "EN" or ?type = "BP").
```

Listing 9. Product identification in the RosettaNet ontology

4 A Semantic RosettaNet Integration Architecture

In this section we outline how RosettaNet collaborations are currently implemented in Service-Oriented Architectures and what additional functionality is required when introducing Semantic Web Services.

4.1 Service-Oriented Architecture (SOA)

Service-Oriented Architecture is an approach to the development of loosely-coupled, protocol-independent and distributed software applications as collections of well-defined autonomous services in a standardised way, enhancing re-usability and interoperability. The SOA paradigm intends to model the business as a collection of services that are available across the enterprise that can be invoked through standard protocols. SOA is often realised through Web Services technologies (Papazoglou & Heuvel, 2007).

The idea of SOA is to use documents as loosely-coupled interfaces and thereby hiding implementation details (Glushko & McGrath, 2005). Therefore, the RosettaNet PIPs form natural services based on standard protocols. For instance, in the PIP 3A4 Purchase Order (PO) process, the seller provides a service that receives a PIP 3A4 PO request messages. After internally checking the PO, the seller sends the PIP 3A4 PO confirmation as the response to the request. The PO message information is extracted and saved to the back-end systems that need to expose some sort of service interfaces to support those interactions. In this collaboration, the standard RosettaNet messages are the common standard protocol.

The current RosettaNet-based integrations often utilise SOA associated products. These products o
ered by many major software companies (e.g. Microsoft BizTalk, TIBCO BusinessWorks, Oracle BPEL Process Manager etc.) support Web Service technologies and B2B standards such as RosettaNet (Medjahed et al., 2003). They have functionality related to design the private and public processes of a collaboration participant and to define the mapping between different schemas. They also contain adapters to popular packaged application to facilitate interoperability with those applications.

4.2 Semantic Service-Oriented Architecture

The notion of a semantic Service-Oriented Architecture (sSOA) (Haller et al., 2005) has been introduced to capture the additional requirements on the execution of Semantic Web Services. In order to avail of the benefits of ontologised RosettaNet messages we propose a light-weight adoption of the WSMX architecture (Haselwanter et al., 2006).

We introduce the following services to a traditional SOA architecture; a knowledge base replacing the service repository in a traditional SOA, ontology adapters ensuring the lifting and lowering of traditional XML Schema based messages to an ontological level and a reasoner service for query answering over the knowledge base, see figure 2.

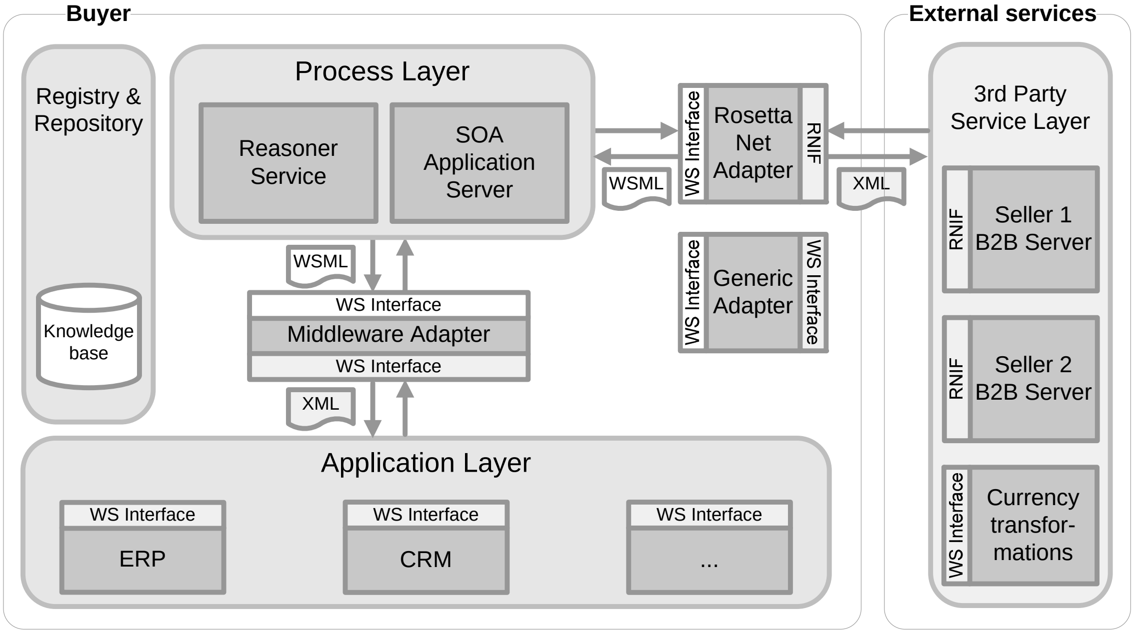

Fig. 2. Overview of the integration architecture

4.2.1 Knowledge base

Similar to traditional SOAs, the knowledge base is a mechanism that handles the persistence of distributed data. However, in an sSOA the data is stored in ontology language documents. In the RosettaNet collaboration these documents include the ontology schemas as well as runtime instances generated by the adapter after the receiving of PIP messages. The knowledge base is used by the reasoner service to perform information retrieval and query-answering functionality.

4.2.2 Reasoner Service

The reasoner is required to perform query answering operations on the knowledge base, including the collaboration instance data during execution. Reasoning is an important functionality throughout the execution process and it is used by the adapters to perform data mediation. It can further be used by a decision support system after and during the execution for the selection of a provider based on the domain specific rules included in the ontology.

The type of reasoner required is dependent on the variant of WSML to be used for the semantic descriptions. The reasoner service offers an abstraction layer (WSML2Reasoner[8]) on top of several reasoners. Depending on the variant used, the queries are passed to the appropriate underlying reasoning engine. Since the current RosettaNet ontology falls into the WSML-Rule variant, we use IRIS[9] and Flora-2 (Yang et al., 2003) for reasoning services.

4.2.3 Adapter Framework

The integration of the Buyer's back-end applications and the B2B standards of each Seller require specific mapping functionality. The role of the adapters is translating the non-WSML data formats to WSML and taking care of B2B standard specific details. The adapters act as the actual service provider for the semantic B2B gateway. The service interface of the adapter is used by the sSOA to invoke the provider functionality instead of the

RosettaNet service endpoint of the partner. Thus, essentially the adapter functionality is registered as a service with the system. Further, the sSOA only operates on the choreography of the adapter (c.f. left part of figure 4), which maps between the choreography of the partner's (c.f. right part of figure 4) e-business environment and the choreography registered with sSOA. The choreography definition is part of the WSMO description of a service and specifies the input and output operations as well as transition rules and constraints on the states in the communication.

RosettaNet adapters The basic functionality of the RosettaNet adapter involves the functionality related to enveloping, encrypting and decrypting and validation of RosettaNet messages. Here, the existing B2B gateway functionality can be used if the organisation already has a product for RNIF communication. Tikkala et al. (2005) present a system with such a functionality, which can handle the RNIF 2.0 related tasks.

When working with sSOA, the mapping rules need to be defined for the runtime phase to lift RosettaNet XML instance messages to the WSML ontology and lower it back to the XML level respectively (c.f. Figure 3).

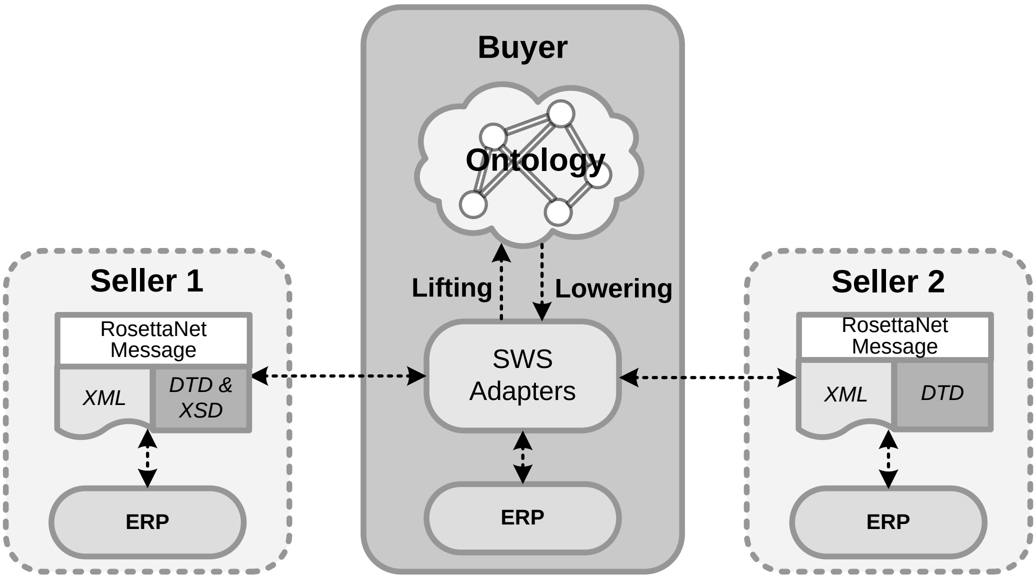

Fig. 3. Lifting/Lowering to/from Domain Ontology

In the scenario mapping rules for PIPs 3A1 and 3A4 are required. We lift the messages to the domain ontology and essentially implement the mediation in the adapter using XSLT stylesheets. Listing 10 contains a snippet of such an example mapping from a DTD-based PIP to WSML. Listing 11 shows the same mappings, but for the XML Schema version PIP. We assume that Seller 1 uses the XML Schema version of RosettaNet PIP 3A4. The mapping lifts the GTIN number to the uniform identification scheme in the ontology. In the lowering of messages, by knowing that a GTIN identifier and company-specific identifiers point to the same product, the mapping can provide an identifier needed by the given partner. As the product information definitions in all DTD and XML Schema based PIPs are similar, these mapping templates can be reused with all the PIPs. With small modification it is easy to create templates for other B2B standards as well.

```
<xsl:for-each select="ProductIdentification/GlobalPartnerClassificationCode">
  instance localUID memberOf productIdentification
    productIdentifier hasValue <xsl:value-of select="."/>
    qualificationAgency hasValue GTIN
</xsl:for-each>

<xsl:for-each select="ProductIdentification/PartnerProductIdentification/">
  instance localUID memberOf productIdentification
    <xsl:for-each select="ProprietaryProductIdentifier">
      productIdentifier hasValue <xsl:value-of select="."/>
    </xsl:for-each>
    <xsl:for-each select="GlobalPartnerClassificationCode">
      qualificationAgency hasValue <xsl:value-of select="."/>
    </xsl:for-each>
</xsl:for-each>
```

Listing 10. DTD-based XSLT mapping

```
<xsl:for-each select="ProductIdentification/GTIN">
  instance localUID memberOf productIdentification
    productIdentifier hasValue <xsl:value-of select="."/>
    qualificationAgency hasValue GTIN
</xsl:for-each>

<xsl:for-each select="ProductIdentification/AlternativeIdentifier/">
  instance localUID memberOf productIdentification
    <xsl:for-each select="Identifier">
      productIdentifier hasValue <xsl:value-of select="."/>
    </xsl:for-each>
    <xsl:for-each select="Authority">
      qualificationAgency hasValue <xsl:value-of select="."/>
    </xsl:for-each>
</xsl:for-each>
```

Listing 11

Listing 11. XML-Schema-based XSLT mapping

Figure 4 shows the RosettaNet adapter execution flow in a PIP process of RFQ request and response messages as described in the motivating example. The execution flow is logically identical with PIP 3A4.

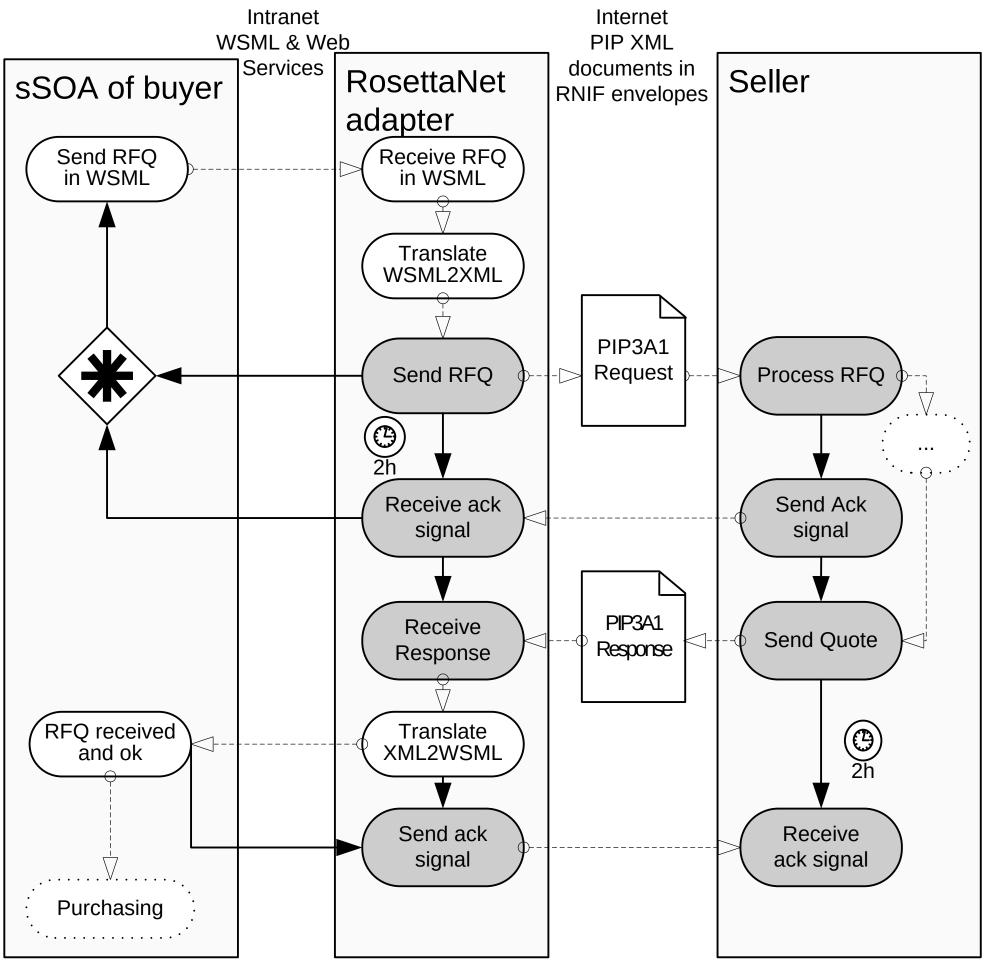

Fig. 4. RosettaNet Adapter

- The sSOA first sends the WSML message to the RosettaNet adapter as a WSML RFQ message.
- The adapter receives the RFQ and translates it to a RosettaNet RFQ XML message.
- The adapter creates an RNIF 2.0 envelope for this message and signs the message using certificates and sends it to the endpoint of a Seller (certificate as well as endpoint

information are implemented in the adapter). As a result, a confirmation that the message has been received is sent back to sSOA. This information is used to start the 24h countdown during which the Seller has time to answer with a quote response.

- The sSOA subsequently expects an acknowledgment message by the Seller in form of an RNIF 2.0 signal message. The acknowledgement needs to arrive in 2 hours or the original RFQ is sent again as indicated in the PIP.
- After receiving the acknowledgment, the adapter is waiting for the Quote response from the Seller.
- The adapter receives the Quote response and translates it using an XSLT script to WSML and sends it to the sSOA to check that the response does not violate the axioms of the ontology. This response is processed in sSOA and sent to the back-end applications. sSOA also forwards an acknowledgment signal indicating that their Quote response was received at the adapter, which sends the acknowledgment signal back to the Seller.

Back-End Adapters Likewise to the RosettaNet adapter, specific adapters to integrate various back-end systems are required. The messages used within the back-end system (e.g. XML or flat-file) have to be mapped to/from the domain ontology. The back-end system adapters are required to perform the lifting/lowering of the internally used messages between the back-end systems and the sSOA.

4.3 Description of the complete message flow with heterogeneous sellers

This section introduces an example runtime behaviour when interacting with two sellers. We describe the execution process and interactions in the sSOA according to the scenario: (1) initiation of a request for specific parts by the back-end systems, (2) requesting quotes from potential sellers for price and availability, (3) selecting the best quote, (4) initiating the Purchase Order process with the selected supplier and returning the details to the back-end system. Figure 5 shows the run-time behaviour of the solution:

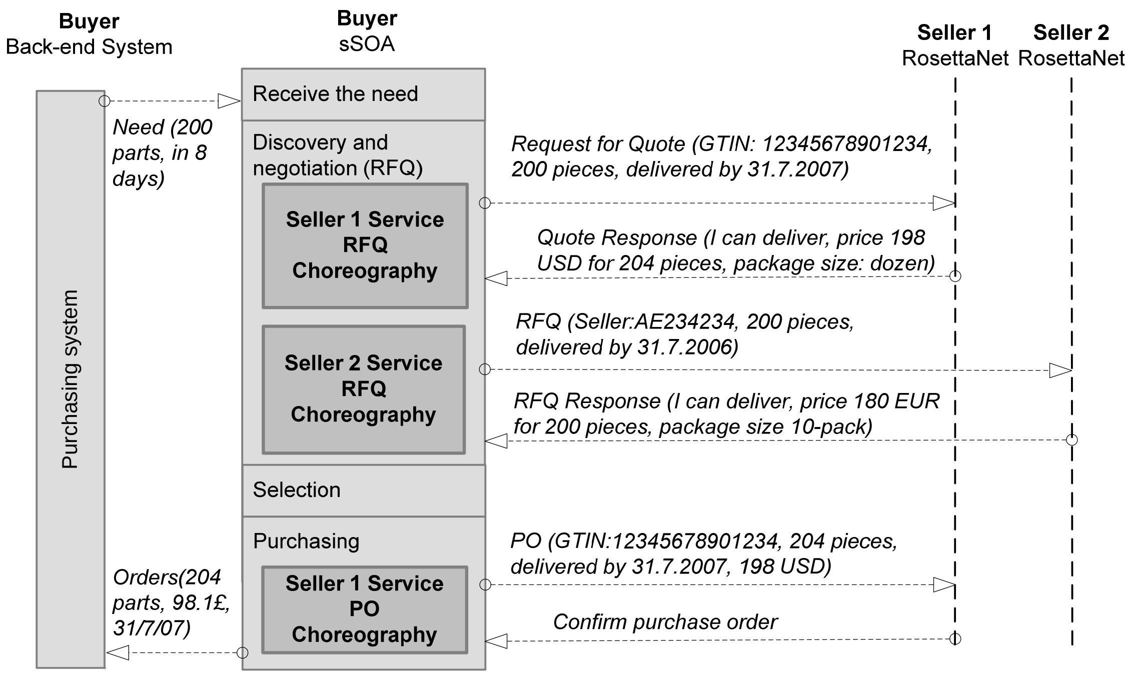

Fig. 5. Runtime behaviour of the proposed solution

Getting the service need The buyer's back-end system issues a request in its proprietary format to the back-end adapter. The request is *to get 200 parts delivered to the plant within 8 days*. The adapter translates the request into a request in WSML, the format required by the sSOA for matching potential services.

Requesting quotes All services in the sSOA repository that match the request are found. In our case the services of Sellers 1 and 2 are discovered as potential suppliers for the requested part. The PIP 3A1 “Request for Quote” choreography (RFQ) is initiated with both Sellers to identify whether they can indeed deliver the required product within the given schedule and to discover their quoted price.

The two sellers offer quotes that differ in packaging sizes, currencies and part identifiers. These heterogeneities in request and response messages are automatically mediated by the RosettaNet adapter, which translates all message exchanges during the collaborative process, as was shown previously in Figure 4. For reference, extracts[10] of the sellers' PIP 3A1 “Quote response messages” in XML and in WSML are shown in listing 12.

```
            <GlobalTransportEventCode>Dock</
                GlobalTransportEventCode>
              <ProductQuantity>204
              </ProductQuantity>
      <GlobalProductUnitOfMeasureCode>dozen
      </GlobalProductUnitOfMeasureCode>
      <SubstituteProductReference>
        <GlobalProductSubstitutionReasonCode>
            Better product
        </GlobalProductSubstitutionReasonCode>
      </SubstituteProductReference>
    <totalPrice>
      <FinancialAmount>
        <GlobalCurrencyCode>USD
        </GlobalCurrencyCode>
        <MonetaryAmount>198
        </MonetaryAmount>
      </FinancialAmount>
    </totalPrice>
```

```
instance QuoteLineItem1 memberOf
    rfq#quoteLineItem
  rfq#globalProductUnitOfMeasurementCode
    hasValue "dozen"
instance quantitySchedule1 memberOf
    core#quantitySchedule
  core#productQuantity hasValue "204"
instance substituteProductReference1 memberOf
    core#substituteProductReference
  core#GlobalProductSubstitutionReasonCode
    hasValue "Better product"
instance totalPrice1 memberOf core#totalPrice
  core#financialAmount
    hasValue FinancialAmountTot
instance FinancialAmountTot memberOf
    core#FinancialAmount
  core#globalCurrencyCode hasValue USD
  core#monetaryAmount hasValue "198"
```

Listing 12. Extract of Seller 1's PIP 3A1 response in XML (left) and WSML (right)

Selection After both sellers provide their quotes and the messages are received and lifted to WSML, the rules defined at design-time ensure that facts are inserted to the knowledge base to obtain homogenous ontology instances. These are used for comparison to select the best supplier for the requested service. In this scenario, the selection is performed according to the cheapest price per unit. To find the cheapest price, a conversion of the different currencies and packaging sizes used in the quotes is performed inside the sSOA through an appropriate currency transformation service and packaging unit transformation service. Since currency rates are volatile, the transformations cannot be expressed in logical axioms; instead, an external Web service is invoked at runtime. In this case, Seller 1 offers the cheaper quote and is therefore selected.

Purchasing The ”purchase order” process (PO) is initiated by the sSOA with the selected Seller 1. The process is similar to the “Request for Quote” process except that the message exchange follows PIP3A4 instead of PIP3A1.

After receiving the PO response from Seller 1, the sSOA performs again the necessary data mediation for product identifiers and currencies; after mediation the translated PO response is returned to the initial back-end system (using a format and structure native to the back-end's interface).

5 Alternatives

Anicic et al. (2006) present how two XML Schema-based automotive standards, AIAG and STAR, are translated from XML to an OWL-based ontology using XSLT. The authors use a two-phase design and runtime approach. Their approach focuses on the conceptual lifting of XML Schemas to OWL.

Brambilla et al. (2006) present a prototype solution based on ontologies designed in the Web Service Modeling language to tackle the factitious requirements defined in the semantic Web Service challenge[11]. The system uses a software engineering approach and utilises a commercial visual modelling tool.

Dogac et al. (2006) present Artemis, a system to semantically enrich Web Services in the healthcare domain. The work discusses how healthcare data defined in domain standards, such as HL7 and CEN, can be formally represented and how these formal representations are used in the execution of semantic Web Services. The presented architecture clearly distinguishes between the functional Web Service interfaces and the application data.

Foxvog & Bussler (2006) describe how EDI X12 can be presented in the WSML, OWL and CycL ontology languages. The work focuses on issues encountered when building a general-purpose B2B ontology.

Khalaf (2007) describes an implementation applying the Business Process Specification Language (BPEL) to RosettaNet PIPs. The objective is to allow the partners to exchange details about their business processes. The evaluation is done by presenting the solution to the RosettaNet board and the BPEL Technical Committee in OASIS. In the test setup, the author deployed multiple servers running different instances representing different companies. The implementation used Web Services instead of RNIF to demonstrate easier agreement of choreography details. PIPs 3A2 and 3A4 were used in the demonstration.

Preist et al. (2005) present a prototype solution covering all phases of a B2B integration life-cycle, starting from discovering potential partners to performing integrations including mediations. The work also addresses issues in translating messages to RDF. The demonstrator uses XML, RDF and OWL-DL. Back-end integration or security aspects are not addressed in this work.

Trastour, Bartolini, & Preist (2003) and Trastour, Preist, & Coleman (2003) augment RosettaNet PIPs with partner-specific DAML+OIL constraints and use agent technologies to automatically propose modifications if the partners use messages differently.

RosettaNet is used in many of the solutions described above, but none of them applies a RosettaNet ontology in combination with external business ontologies to automatically homogenise the message content from different providers and select services based on the most competitive offer. All solutions are still use case based and lack comprehensive evaluations.

6 Cost and Benefits

The current long and expensive setup of B2B integrations has lead to business models with simple public processes in which long term rigid partnerships are established between organisations. In RosettaNet collaborations there is often no competition in the request for quotes, as usually the default partner is selected directly for purchasing using long-term contracts (Kotinurmi et al., 2006). This is partly due to the overhead to manage multiple partner specific quoting and purchasing integrations.

The presented solution shows, how heterogeneities, caused by the introduction of new partners to the supply chain, are reduced and how more competitive arrangements can be organised. The quoting and purchasing scenario highlights some of the problems currently observed in RosettaNet collaborations. For example, accepting suppliers from different countries causes heterogeneities, as the partners are likely to use different currencies, different measurement units or different packaging units. The current DTD-based PIPs even enforce organisations to implement additional validation means. However, even with the introduction of XSD based PIPs, one can represent the same information within message instances in different ways. For example, some PIPs allow time durations to be specified either using “starting time and duration” or “starting time and ending time”. Such freedom in the specification requires both partners to agree on the particular representation and its implementation. Any changes to the way information is represented according to the schema causes additional costs in the setup.

The benefits of resolving such heterogeneities for the buyer result from decreased costs of purchasing as the best value deals can be selected on the basis of the best quotes. The sellers benefit from being able to integrate to multiply buyers without making potentially costly changes to their current interfaces. Only declarative rules defining how to interpret incompatible message instances have to be added. Contrary to traditional B2B collaborations where functions are developed on a point-to-point basis, the explicitly modelled relations between concepts in RosettaNet ontology help to resolve data heterogeneities globally and are thus reusable in every partner integration. The relations between the elements in the RosettaNet specification are not specified by the current schemas as they are not expressive enough to present such relationships.

Being able to handle the heterogeneities caused by the freedom in the specifications is particularly important for companies that cannot dictate the use of B2B standards to their partners. Additional partners are introduced more easily to the supply chain and the possibility to have more competitive partnerships benefits the buyers. The solutions provided have potential use in a significant portion of all RosettaNet PIPs, of which roughly half contain currency and measurement unit information. According to McComb (2004), more than half of the 300 billion dollars annually spent on systems integration is spent on resolving semantic issues.

The costs associated with the presented solution are characterised by two distinctive features; high ontology engineering costs, but low subsequent setup costs. The high costs associated with the ontology engineering can be compared with the development of the RosettaNet framework. Highly skilled knowledge engineers are required to ontologise the full semantics implicit to the current RosettaNet specifications. Following a bottom-up approach the solution presented takes the existing schema definitions and gradually introduces formal semantics for different kind of heterogeneity sources. It is a time-consuming knowledge engineering task to encompass all RosettaNet business documents in the ontology, but it is a one-time effort. We presented an approach to automatically derive a core RosettaNet ontology in Haller et al. (2008). Similar ontologisation efforts are undertaken in multiple domains, such as the CyC ontology[12] for general knowledge, the Geonames ontology[13] for geospatial information, uniProt[14] for protein sequences and the Gene Ontology[15] describing gene and gene product attributes in any organism. As discussed by Preist et al. (2005), the use of dynamic integration via semantic descriptions and generic e-business ontologies is expected to become an important industrial technique in the near future; an example of such a generic e-business ontology is the eClassOWL[16] ontology, an ontologised taxonomy of product and service categories (Hepp, 2005a,b; Hepp et al., 2005).

Similarly to traditional B2B integrations (Bussler, 2003) adapters are required for the communication on the application layer as well as the service layer. The behaviour of the RosettaNet adapter concerning RNIF is identical in all RosettaNet PIPs and thus needs to be defined just once. Adapters for different partners at the service layer in an ontologically enhanced RosettaNet solution have the advantage over schema matching approaches that the semantic relations are reusable. Instead of partner-specific point-to-point XSLT transformations, semantic relations are specified only once, but then reused in many collaborations. Multiple approaches exist to automatically (Aumueller et al., 2005; Hu et al., 2006; Tang et al., 2006) or semi-automatically (Maedche et al., 2002; Mocan & Cimpian, 2007; Noy & Musen, 2000) derive such semantic relations between two ontologies. However, the mappings are easier to create, even if they have to be done manually, since the schema (ontology) makes all data semantics explicit.

7 Risk Assessment

Several publications on semantic integrations (Anicic et al., 2006; Cabral et al., 2006; Preist et al., 2005; Trastour, Preist, & Coleman, 2003; Vitvar et al., 2007) have shown that conceptually, B2B integrations can benefit from semantic technologies. However, all documented implementations are prototypical and all mapping and mediation capabilities are on a use case basis. Recently, more standardised evaluation procedures have been initiated in the context of the Web Service[17] and semantic Web Service[18] challenges. These initiatives are a first step towards a unified and realistic evaluation framework.

In terms of a migration strategy for existing infrastructure and processes, the solution presented in this chapter does not mandate changes to the infrastructure of the business partners. Rather, the solution builds on top of existing B2B standards: business partners can still utilise their existing infrastructure, while the buyer profits from applying semantic Web technologies.

In terms of performance and scalability issues, these have not yet been addressed in current evaluations. Especially the lifting and lowering of message instances to their ontological representations in the adapters requires additional computational resources as compared to existing solutions; since production PIP instances have been reported to be even hundreds of megabytes large (Damodaran, 2004), scalability and performance are obvious concerns in such settings. Further testing is needed to evaluate scalability and performance, as for example done with RosettaNet technologies (Tikkala et al., 2005). However, in the presented architecture, reasoning is only performed at design time, since only one partner uses message ontologies. Thus, no runtime ontology mediation is required and instead all mapping logic is coded into the back-end and RosettaNet adapters. Even current SOA implementations are said to “think SOA and implement high-performance transaction system” to address similar concerns with basic Web Service technologies (Bussler, 2007).

The strengths, weaknesses, opportunities and threats of the Ontologically-enhanced RosettaNet solution are analysed below.

Strengths The main strengths are the possibility to increase competition in the supply chain, by simplifying runtime integration with multiple service providers, the lowered integration costs since many manual integration and alignment tasks (e.g. measurement unit translation and currency alignment) can be automated; and the possibility to automatically validate incoming business messages against ontologised business constraints (which are currently only informally documented at best).

Weaknesses The main weaknesses are the required skills in ontological modelling, which might not be available in organisations given the young age of semantic technologies; the computational overhead in lifting and lowering XML messages, and the required investment in the initial development of domain ontologies. On the other hand, like any other standardisation effort, the costs of ontology development can be shared by the entire business domain and development costs can be relatively low due to a high level of potential ontology reuse.

Opportunities The main opportunities are for reuse of existing top-level and e-commerce ontologies such as Cyc and eClassOWL. Reuse is central to ontology development and offers significant business opportunities and savings through accumulated network effects. Further, the explicit formal semantics of ontologised RosettaNet messages offer opportunities for automated testing of integration solutions, which currently have to be verified manually. Finally, as in any innovative technology, an opportunity exists for early adopters to master the solution and reuse it across other domains, or even to become solution providers for other organisations in the B2B integration marketplace.

Threats The main threats are the highly volatile environment of organisations which may hamper technology that requires some setup time, which may not be tolerated in such volatile environments. The uptake of semantic technologies is still limited and the tool support for semantic technologies is immature.

Overall, the present approach is best suited for organisations that encounter multiple heterogeneity issues and cannot dictate their partners the B2B integration details. If an organisation is able to dictate the B2B integration details onto their partners, the benefits of semantically enriched messages are limited since such an organisation can achieve homogeneity through its mandate. Similarly, organisations with only a limited number of partner relationships the introduction of ontologised RosettaNet messages might be an overhead.

8 Future Trends and Additional reading

Current approaches to Enterprise Application Integration (Erasala et al., 2003; Linthicum, 1999) and B2B integration (Bussler, 2003) are still based on static binding to services and partners. One of the major trends in enterprise information system development is model-driven design (Brown, 2004). The promise is to create business-driven IT solutions quickly and reduce the gap between business experts and technology experts. SOA and Business Process Management standards are one step towards this goal. Ultimately, the models in an sSOA should span from formal data representations to formal business process models. Different abstraction levels should be supported such that the business analyst can use a modelling environment on a high abstraction level, but still the resulting process model can be used by the implementer as a scaffolding model. However, currently data models are still disconnected from process models and business process models on different abstraction levels, such as Solution Maps and Business Process Execution Language (Andrews et al., 2003) based models, are disconnected. Several recent research projects[19] and publications (Haller et al., 2006; Hepp & Roman, 2007) are trying to tackle the issue of relating different abstraction levels of business process modelling.

Other research topics related to formal process models in sSOA are the automatic composition of services and processes (Milanovic & Malek, 2004). Based on a formal model it is possible to generate a plan (process model) automatically. Most approaches to service composition are related to AI planning (McIlraith & Son, 2002; Wu et al., 2003), deductive theorem proving (Rao et al., 2006) and model checking (Berardi et al., 2005; Bultan et al., 2003). The general assumption is that every Web Service can be specified by its preconditions and effects in the planning context.

An important step towards a dynamic integration, both within and across enterprise boundaries, is a scalable and ultimately fully mechanised approach to service discovery. Automatically locating services to achieve a certain business goal can considerably reduce the development costs of B2B integrations. Providers can be dynamically selected based on the service they provide and on the properties associated with the services, such as its cost, the trust relation, security, etc. For instance, in the context of our motivating example the discovery based on the product's classification category enables a meaningful way to compare substitutable products with the initially requested items.

The solution presented in this chapter does not avail of dynamic service selection. However, as a promising research direction multiple e orts are dedicated to solve its challenges (Keller et al., 2005; Ran, 2003; Sycara et al., 2004).

9 Conclusions

The scenario discussed in this chapter on quoting and purchasing highlights the problems currently observed in RosettaNet collaborations. RosettaNet represents the current state-of-the-art in B2B integration, and its usage of XML is an improvement compared to more traditional integration solutions. However, setting up integrations using RosettaNet still requires considerable manual effort due to the heterogeneity that arises from the interpretation freedom in the RosettaNet specification.

Specifically, RosettaNet integrations still suffer from interoperability challenges in aligning the business processes of the partners, message heterogeneity due to the lack of expressive power in the schemas, and ambiguity in the type definitions, again due to the low expressivity of the schema languages used in RosettaNet.

As a consequence of the required manual integration effort, current business models in B2B integration are still simple and integration of new partners is difficult, limiting the possibility of introducing more competition in the supply chain. For example, interacting with suppliers from different countries introduces heterogeneities, since the suppliers are likely to use different currencies, different measurement units or different packaging unit. Due to the current type definitions in RosettaNet, manual effort is needed to overcome these heterogeneities.

The ontologically-enhanced RosettaNet solution presented in this chapter helps tackling heterogeneities in RosettaNet interactions. The solution relies upon a formalised RosettaNet ontology to capture otherwise implicit knowledge and rules to resolve data heterogeneities. The ontology captures definitional facts, such as the relation between measurement units or product category memberships. Such information is currently not modelled in RosettaNet. The solutions provided in this chapter have potential use in significant portions of all RosettaNet Partner Interface Processes (PIPs). Furthermore, ontologising RosettaNet messages provides support for better discovery, selection and composition of PIPs and PIP parts for other situations.

By automatically resolving such heterogeneities, the buyer gains benefits through decreased costs of purchasing as the best value deals can be selected based on the best quotes. The sellers benefit from being able to easier integrate to the buyer without having to make potentially costly changes to their current integration interfaces. The ontologically enhanced RosettaNet solution enables scalable, fast and easy to monitor processes eliminating unnecessary repetitive manual work and enabling people to concentrate on value-adding activities. Furthermore, many testing tasks can be avoided, since formal algorithms can check the quality of the processes such as the existence of data transformation from the buyer's schemas into the seller's schemas.

References

Aalst, W. M. van der, & Kumar, A. (2003). XML-Based Schema Definition for Support of Interorganizational Workflow. *Information Systems Research, 14* (1), 23-46.

Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., et al. (2003, May). *Business Process Execution Language for Web Services, v1.1*.

Anicic, N., Ivezic, N., & Jones, A. (2006). An Architecture for Semantic Enterprise Application Integration Standards. In D. Konstantas, J.-P. Bourrires, M. Lonard, & N. Boudjlida (Eds.), (pp. 25-34). London, UK: Springer.

Aumueller, D., Do, H. H., Massmann, S., & Rahm, E. (2005). Schema and ontology matching with COMA++. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 906-908). Baltimore, Maryland, USA.

Battle, S., Bernstein, A., Boley, H., Grosof, B., Gruninger, M., Hull, R., et al. (2005). *Semantic Web Services Framework (SWSF) Overview* (Member Submission). W3C. (Available from: http://www.w3.org/Submission/SWSF/)

Berardi, D., Calvanese, D., Giacomo, G. D., Lenzerini, M., & Mecella, M. (2005). Automatic Service Composition Based on Behavioral Descriptions. *Int. J. Cooperative Inf. Syst., 14* (4), 333-376.

Brambilla, M., Celino, I., Ceri, S., Cerizza, D., Valle, E. D., & Facca, F. M. (2006). A Software Engineering Approach to Design and Development of Semantic Web Service Applications. In *Proceedings of the 5th International Semantic Web Conference*. Athens, GA, USA: Springer.

Brown, A. (2004). An Introduction *to Model Driven ArchitecturePart I: MDA and Todays Systems* (Tech. Rep.). IBM. (Available from: http://www.ibm.com/developerworks/rational/library/3100.html)

Bruijn, J. de, Bussler, C., Domingue, J., Fensel, D., Hepp, M., Keller, U., et al. (2005). *Web Service Modeling Ontology (WSMO)* (Member Submission). W3C. (Available from: http://www.w3.org/Submission/WSMO/)

Brunnermeier, S. B., & Martin, S. A. (2002). Interoperability costs in the US automotive supply chain. *Supply Chain Management: An International Journal, 7* (2), 71-82.

Bultan, T., Fu, X., Hull, R., & Su, J. (2003). Conversation specification: a new approach to design and analysis of e-service composition. In *Proceedings of the 12th international conference on World Wide Web* (pp. 403-410). New York, NY, USA: ACM Press.

Bussler, C. (2003). *B2B Integration: Concepts and Architecture*. Springer.

Bussler, C. (2007). The Fractal Nature of Web Services. *IEEE Computer, 40* (3), 93-95.

Bussler, C., Fensel, D., & Maedche, A. (2002). A Conceptual Architecture for Semantic Web Enabled Web Services. *SIGMOD Record, 31* (4), 24-29. ACM Press.

Cabral, L., Domingue, J., Galizia, S., Gugliotta, A., Tanasescu, V., Pedrinaci, C., et al. (2006). IRS-III: A Broker for Semantic Web Services Based Applications. In *Proceedings of the 5th International Semantic Web Conference.* (pp. 201-214). Athens, GA, USA:Springer.

Clark, J., Casanave, C., Kanaskie, K., Harvey, B., Clark, J., Smith, N., et al. (2001). *ebXML Business Process Specification Schema (Version 1.01).* ebXML. (Available from: http://www.ebxml.org/specs/ebBPSS.pdf/)

Damodaran, S. (2004). B2B integration over the Internet with XML: RosettaNet successes and challenges. In *Proceedings of the 13th International World Wide Web Conference on Alternate track papers & posters.* (pp. 188-195). New York, NY, USA: ACM Press.

Damodaran, S. (2005). RosettaNet: Adoption Brings New Problems, New Solutions. In *Proceedings of the XML 2005 Conference.* (pp. 1-14). Atlanta, USA: IDE Alliance.

Dogac, A., Laleci, G. B., Kirbas, S., Kabak, Y., Sinir, S. S., Yildiz, A., et al. (2006). Artemis: Deploying semantically enriched Web services in the healthcare domain. *Information Systems, 31* (4-5), 321-339.

Erasala, N., Yen, D. C., & Rajkumar, T. M. (2003). Enterprise Application Integration in the electronic commerce world. *Computer Standards & Interfaces, 25* (2), 69-82.

Fensel, D. (2003). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer.

Foxvog, D., & Bussler, C. (2006). Ontologizing EDI Semantics. In *Proceedings of the Workshop on Ontologising Industrial Standards*. (pp. 301-311). Tucson, AZ, USA: Springer.

Glushko, R. J., & McGrath, T. (2005). Document engineering: analyzing and designing the semantics of business service networks. In *Proceedings of the IEEE EEE05 international workshop on business services networks*. (pp. 9-15). Piscataway, NJ, USA: IEEE Press.

Grosof, B. N., Horrocks, I., Volz, R., & Decker, S. (2003). Description logic programs: combining logic programs with description logic. In *Proceedings of the 12th International World Wide Web Conference* (pp. 48-57). New York, NY, USA: ACM Press.

Gruber, T. R. (1993). Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In N. Guarino & R. Poli (Eds.), *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Deventer, The Netherlands: Kluwer Academic Publishers.

Haller, A., Cimpian, E., Mocan, A., Oren, E., & Bussler, C. (2005). WSMX - A Semantic Service-Oriented Architecture. *In Proceedings of the 3rd International Conference on Web Services,* (pp. 321-328). Orlando, Florida, USA: IEEE Computer Society.

Haller, A., Gontarczyk, J., & Kotinurmi, P. (2008). Towards a complete SCM Ontology - The Case of ontologising RosettaNet. In Proceedings of the 23rd ACM symposium on applied computing (sac). Fortaleza, Ceara, Brazil. (to appear)

Haller, A., Kotinurmi, P., Vitvar, T., & Oren, E. (2007). Handling heterogeneity in RosettaNet messages. In Y. Cho, R. L. Wainwright, H. Haddad, S. Y. Shin, & Y. W. Koo (Eds.), *SAC*, (pp..1368-1374). ACM.

Haller, A., Oren, E., & Kotinurmi, P. (2006). m3po: An ontology to relate choreographies to workflow models. In *Proceedings of the 3rd international conference on services computing*, (pp. 19-27). Chicago, Illinois, USA: IEEE Computer Society.

Haselwanter, T., Kotinurmi, P., Moran, M., Vitvar, T., & Zaremba, M. (2006). WSMX: A Semantic Service Oriented Middleware for B2B Integration. In *Proceedings of the International Conference on Service-Oriented Computing*, (pp. 477-483). Springer.

Hepp, M. (2005a). eClassOWL: A fully-fledged products and services ontology in OWL. In *Proceedings of the International Semantic Web Conference* (ISWC).

Hepp, M. (2005b). A methodology for deriving OWL ontologies from products and services categorization standards. In *Proceedings of the 13th European Conference on Information Systems* (ECIS2005), (pp. 1-12).

Hepp, M., Leukel, J., & Schmitz, V. (2005). A Quantitative Analysis of eCl@ss, UNSPSC, eOTD, and RNTD Content, Coverage, and Maintenance. In *Proceedings of the IEEE ICEBE 2005 Conference*, (pp. 572-581).

Hepp, M., & Roman, D. (2007). An Ontology Framework for Semantic Business Process Management. In *Proceedings of the 8th International Conference Wirtschaftsinformatik* 2007.

Hu, W., Cheng, G., Zheng, D., Zhong, X., & Qu, Y. (2006). The Results of Falcon-AO in the OAEI 2006 Campaign. In *Proceedings of the 1st International Workshop on Ontology Matching*. Athens, GA, USA.

Keller, U., Lara, R., Lausen, H., Polleres, A., & Fensel, D. (2005). Automatic Location of Services. In *Proceedings of the 2nd European Semantic Web Conference*, (pp. 1-16). Heraklion, Crete, Greece: Springer.

Khalaf, R. (2007). From RosettaNet PIPs to BPEL processes: A three level approach for business protocols. *Data and Knowledge Engineering, 61* (1), 23-38.

Kotinurmi, P., Vitvar, T., Haller, A., Richardson, R., & Boran, A. (2006). Semantic Web Services Enabled B2B Integration. In J. Lee, J. Shim, S. goo Lee, C. Bussler, & S. S. Y. Shim (Eds.), *DEECS* (pp. 209-223). Springer.

Linthicum, D. (1999). *Enterprise Application Integration*. Reading, MA: Addision-Wesley Longman.

Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). MAFRA - A MApping FRAmework for Distributed Ontologies. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, (pp. 235-250). Springer.

Martin, D., et al. (2004). *OWL-S: Semantic Markup for Web Services* (Member Submission). W3C. (Available from: http://www.w3.org/Submission/OWL-S/)

McComb, D. (2004). *Semantics in Business Systems: The Savvy Manager's Guide*. San Francisco: Morgan Kaufmann.

McIlraith, S. A., & Son, T. C. (2002). Adapting Golog for Composition of Semantic Web Services. In *Proceedings of the 8th International Conference on Principles and Knowledge Representation and Reasoning*. Toulouse, France.

McIlraith, S. A., Son, T. C., & Zeng, H. (2001). Semantic Web Services. *IEEE Intelligent Systems, 16* (2), 46-53.

Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A. H. H., & Elmagarmid, A. K. (2003). Business-to-business interactions: issues and enabling technologies. *VLDB Journal, 12* (1), 59-85.

Milanovic, N., & Malek, M. (2004). Current Solutions for Web Service Composition. *IEEE Internet Computing, 8* (6).

Mocan, A., & Cimpian, E. (2007). An Ontology-Based Data Mediation Framework for Semantic Environments. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 3 (2).

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on formal ontology in information systems*, (pp. 2-9). New York, NY, USA.

Noy, N. F., & Musen, M. A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the 7th National Conference on Artificial Intelligence*, (pp. 450-455). Austin, Texas, USA.

Nurmilaakso, J.-M., & Kotinurmi, P. (2004). A Review of XML-based Supply-Chain Integration. *Production Planning and Control, 15* (6), 608-621.

Papazoglou, M. P., & Heuvel, W.-J. van den. (2007). Service oriented architectures: approaches, technologies and research issues. *VLDB Journal, 16* (3), 389-415.

Preist, C., Cuadrado, J. E., Battle, S., Williams, S., & Grimm, S. (2005). Automated Business-to-Business Integration of a Logistics Supply Chain using SemanticWeb Services Technology. In *Proceedings of 4th International Semantic Web Conference*, (pp. 987-1001). Springer.

Ran, S. (2003). A model for web services discovery with QoS. *SIGecom Exch., 4* (1), 1-10.

Rao, J., Kungas, P., & Matskin, M. (2006). Composition of semantic web services using linear logic theorem proving. *Information Systems, 31* (4), 340-360.

Roman, D., Keller, U., Lausen, H., Bruijn, J. de, Lara, R., Stollberg, M., et al. (2005). Web Service Modeling Ontology. *Applied Ontologies, 1* (1), 77-106.

Shim, S. S. Y., Pendyala, V. S., Sundaram, M., & Gao, J. Z. (2000). Business-to-Business E-Commerce Frameworks. *IEEE Computer, 33* (10), 40{47.

Sycara, K. P., Paolucci, M., Soudry, J., & Srinivasan, N. (2004). Dynamic Discovery and Coordination of Agent-Based Semantic Web Services. *IEEE Internet Computing, 8* (3), pp. 66-73.

Tang, J., Li, J., Liang, B., Huang, X., Li, Y., & Wang, K. (2006). Using Bayesian decision for ontology mapping. *Journal of Web Semantics, 4* (4), 243-262.

Tikkala, J., Kotinurmi, P., & Soininen, T. (2005). Implementing a RosettaNet Business-to- Business Integration Platform Using J2EE and Web Services. In *Proceedings of the 7th IEEE International Conference on E-Commerce Technology* , (pp. 553-558). IEEE Computer Society.

Trastour, D., Bartolini, C., & Preist, C. (2003). Semantic Web support for the business-to-business e-commerce pre-contractual lifecycle. *Computer Networks, 42* (5), 661-673.

Trastour, D., Preist, C., & Coleman, D. (2003). Using Semantic Web Technology to Enhance Current Business-to-Business Integration Approaches. In *Proceedings of the 7th International Enterprise Distributed Object Computing Conference*, (pp. 222-231). IEEE Computer Society.

Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review, 11* (2), 93-155.

Vitvar, T., Mocan, A., Kerrigan, M., Zaremba, M., Zaremba, M., Moran, M., et al. (2007). Semantically-enabled service oriented architecture : concepts, technology and application. *Service Oriented Computing and Applications, 2* (2), 129-154.

Wu, D., Parsia, B., Sirin, E., Hendler, J. A., & Nau, D. S. (2003). Automating DAML-Sweb Services Composition Using SHOP2. In *Proceedings of the 2nd International Semantic Web Conference*, pp. 195-210. Sanibel Island, FL, USA: Springer.

Yang, G., Kifer, M., & Zhao, C. (2003). Flora-2: A rule-based knowledge representation and inference infrastructure for the semantic web. In *Proceedings of the Coopis, doa, and odbase - otm confederated international conferences, On the move to meaningful internet systems 2003*. Catania, Sicily, Italy.

Footnotes

[1] http://www.rosettanet.org/

[2] http://www.openapplications.org/

[3] http://www.ebxml.org/

[4] http://www.oasis-open.org/committees/ubl/

[5] http://www.bpmn.org/

[6] For understanding the syntax, see http://www.wsmo.org/wsml/wsml-syntax

[7] See http://ontologies.deri.org/eclass/5.1/#C_AAA374002

[8] See http://tools.deri.org/wsml2reasoner/

[9] http://iris-reasoner.org/

[10] The full examples are available at http://www.m3pe.org/ontologies/rosettaNet/

[11] http://sws-challenge.org/

[12] http://www.opencyc.org/

[13] http://www.geonames.org/ontology/

[14] http://dev.isb-sib.ch/projects/uniprot-rdf/

[15] http://www.geneontology.org/

[16] http://www.heppnetz.org/eclassowl/

[17] http://ws-challenge.org/

[18] http://sws-challenge.org/

[19] http://www.ip-super.org/, http://www.m3pe.org/

Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. SFI/04/BR/CS0694 and by the Finnish Funding Agency for Technology and Innovation. The authors wish to thank Dr. Tomas Vitvar for participating in the research and commenting the early versions.

Author biographies

Paavo Kotinurmi is a researcher and lecturer at the Software Business and Engineering Institute, Helsinki University of Technology. Paavo Kotinurmi received his Doctoral degree in Computer Science in 2007. His Master of Science degree was completed in 2001. The Ph.D. topic was applying XML-based standards to solve practical B2B integration problems. In that area, also semantic Web Services technologies have been applied to solve integration problems. As part of that, Dr. Kotinurmi spent nine months in DERI Galway as a visiting researcher. His current research interests include combining XML-based B2B standards, Service-Oriented Architectures and semantic technologies. His current research targets helping companies and healthcare organisations to setup Service-Oriented Architectures utilising B2B standards, ontologies and semantic technologies where appropriate.

Armin Haller is Research Assistant in the Digital Enterprise Research institute (DERI) and PhD student at the National University Ireland, Galway. Until recently he was co-managing the m3pe project to design a process ontology to formally capture different process execution models. Currently he is involved in the EU funded SUPER project, which aims to extend existing generic models for the representation of processes with semantic information capturing all aspects of system behaviour. His research interests are Business Process Integration and semantic Web Services, where he published some 20 scientific articles in journals, conferences and workshops and co-edited the Business Process Management Workshops Proceedings in 2005. He is the main developer of m3po, a Business Process Integration ontology, semantic Rosettanet, a document standard ontology and has been working in the WSMX initiative, developing the first semantic Service Oriented Architecture.

Eyal Oren is a researcher at the Network Institute in the Computer Science and Communication Sciences departments of the Vrije Universiteit Amsterdam. Eyal Oren received a Ph.D. in Computer Science from the National University of Ireland, Galway for his thesis on algorithms and components for application development on the Semantic Web. He also holds a M.Sc. degree from the Delft University of Technology in the Netherlands. His research is concerned with techniques for large-scale manipulation and analysis of Semantic Web data and with semantics in workflow management and process management. As part of his research, Dr. Oren has published around thirty articles in international conferences and journals and has spent several months as visiting scholar at Stanford University and at the University of Innsbruck. He now works on information extraction, knowledge representation and formal reasoning techniques for automatic analysis of political news texts.