

Towards an Ontology for Automating Collaborative Business Processes

Bahareh Rahmanzadeh Heravi, David Bell, Mark Lycett
Department of Information Systems and Computing
Brunel University
Uxbridge, London, UB8 3PH, UK
{Bahareh.Heravi, David.Bell, Mark.Lycett}@brunel.ac.uk

Stephen D. Green
Document Engineering Services
Cambridgeshire, CB6 2AP, UK
stephen.green@documentengineeringservices.com

Abstract-Automation of business transactions between trading partners is an important factor in today's global business. XML based E-Business standards are developed to provide a shared understanding on what information to share, when and how between trading partners. However these standards can only capture the syntax of the transactions and not the semantics. This paper presents an ontology for ebXML Business Process Specification Schema (ebBP), with the aim of empowering the capture and sharing semantics embedded within B2B processes, enabling knowledge deduction and reasoning over this shared knowledge. The ebBP ontology presented covers both syntax included in ebBP XML Schema and the informal semantics of the ebBP specification and is fundamentally different to an automatic transformation of XML to OWL. This ontology is evaluated against a set of competency questions, using a publicly available ordering process. This paper demonstrates how semantic web technologies can be utilised in order to improve standards-based interoperability between trading partners.

Keywords-B2B Process, Public Process, Collaborative Process, Ontology, Semantic Web, Semantic Business Process, Process Interoperability, Process Automation, ebBP, ebXML

I. INTRODUCTION

Companies now operate in a global business environment where business values and competitive advantage lie beyond the boundaries of any one enterprise. In such an environment, companies need to interact with different trading partners and, in order to do so, their business processes need to be understood and aligned across organisational boundaries

Business process standards seek to provide a shared understanding and agreement on what information to share, when and how among trading partners. One of these standards is the ebXML Business Process Specification Schema (ebBP), a Business to Business (B2B) process standard, standardised by OASIS (Organization for the Advancement of Structured Information Standards) [14]. EbBP's focus is on public processes and documents exchanged between trading partners in B2B transactions. However, despite the fact that XML based standards, such as ebBP, are designed to provide a common language between their users, XML can only cover syntax and not the semantics of transactions. Therefore there is a need for a semantic based approach for capturing and sharing the semantics of B2B processes between trading partners.

Ontologies are considered as an appropriate means for data and information integration. Firat, Madnick & Manola [2] state that applying ontologies in practical semantic interoperability problems has proven to reduce the amount of work needed to agree on a shared model based on the

assumptions made by different parties. Ontologies can capture the definitions and interrelationships of concepts in a variety of domains [19] and enable reasoning and knowledge deduction. This facilitates transparent flow of semantically enriched information and knowledge in order to enhance B2B collaborations [18].

This paper provides an ontology for ebBP, enabling capturing and sharing semantics embedded in B2B processes and knowledge deduction and reasoning over the shared knowledge. The ebBP ontology¹ presented in this paper, not only covers the syntax of the ebBP XML schema but also provides facilities for covering informal semantics embedded in the textual specification. The ebBP ontology is fundamentally different from automatic transformation of XML to OWL since the automatic transformation cannot cover the semantics embedded in both the schema and the textual specifications.

The remainder of this paper is structured as follows: Section II provides a background on B2B Business Processes and the role of ebBP in B2B transactions. Section III discusses the importance of ontologies and semantic web technologies for B2B process interoperation, followed by Section IV which reviews the related work. In Section V the ebBP ontology is presented and evaluated through presentation of an ebBP process instance and using DL Queries to answer a set of competency questions. Section VI concludes the paper.

II. B2B PROCESSES AND EBBP

A powerful component of B2B e-Business lies in the automation of business transactions using electronic documents and other electronic information. Legal aspects of such business transactions require careful control and specification at a technical level and at a contractual level. For example, there may be a time limit to a transaction after which it becomes void. This agreement of terms is essential to successful electronic automation of business transactions. Parties engaging in the automated business process need to apply the same terms and conditions so that a status such as a contract existing or having been backed out is known equally to each party. Then each party's system knows what is expected of it at any particular time. The business process and its documents and signals are defined such that each party's understanding is the same. There is a subset of the transactions and flow of each collaboration which need to be equally visible to each party. Other parts need not be visible and can be regarded as 'private' or 'internal'. B2B

¹ <http://www.semantic-b2bi.eu/ontologies>

processes, also referred to as ‘collaborative’, ‘inter-organisational’, or ‘public’ processes are those focusing on the interactions between different partners and are not concerned about internal processes. These shared aspects need to be specified using principles essential to automation^[12] so that all systems involved have the same understanding of the state of those collaborations at certain stages in the process.

The public aspects require a shared definition. When transactions in the collaboration stages of the process are automated there are business process specification languages which can be used to clearly define the public aspects of those process steps. One such language is OASIS ebXML Business Process Specification Schema or ebBP [14] which has the benefit that it has been specifically designed for defining the public aspects of an e-business automated collaboration. The parties only need to agree on certain aspects of their transactions for the automation to be successful and the ebBP language is optimised for these. The fact that ebBP is a royalty-free, open standard adds to its usefulness for defining collaborations and transactions in a way which may be correctly understood by each party involved in the business processes.

EbBP is one of five components of ebXML (Electronic Business using eXtensible Markup Language) framework. EbXML is a modular suite of XML based specifications, sponsored by OASIS and UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business). EbXML’s mission is to provide an open, XML based infrastructure that enables the global use of electronic business information in an interoperable, secure, and consistent manner. The other four components of the ebXML framework are as follows:

- *ebXML Core Components*, which provide basic and reusable building blocks for Business Documents.
- *ebXML Registry/Repository (ebReg/Rep)*, which are as follows. EbXML Repository manages and maintains the shared information as objects in a repository. EbXML registry is an interface for accessing and discovering shared business semantics.
- *Collaboration Protocol Profiles and Agreements (CPP/A)*, which are as follows. The CPP describes the specific capabilities that a trading partner supports. A CPA is a document that represents the intersection of two CPP’s and is mutually agreed upon by both trading partners.
- *ebXML Messaging Service (ebMS)*, which is designed for the secure, reliable exchange of e-Business information.

The ebXML framework is designed in a way that specifications of each component can be used independently, composed as desired, or integrated with other evolving technologies^[14].

The Focus of this paper is on Business Processes standards and therefore on the ebBP as a self contained unit which may be used in conjunction with other specifications or technologies.

III. ONTOLOGIES AND B2B PROCESSES

There are different types of specification needed in e-business, with diagrams, XML artifacts such as a schema, tables and flat files such as code-lists. The workflows downstream from production of a specification may require that the logic be restated using precise, formal expressions that can be parsed and interpreted by software and by software engineers not expert in the e-business domain. This does not always involve semantic interpretation. It is sometimes sufficient to create logical rules related only to the specified syntax and structure. However, when a formal expression of the semantic information is required ontology provides an important tool. Converting to an ontology makes the semantics accessible to automated processing and to engineers not expert in the e-business domain.

An ontology is “an explicit specification of a conceptualisation” [6, p.1]. Ontologies provide a formal description of concepts and their relationships within a domain [23], which result in a shared understanding of a domain. This understanding, expressed using formal logic, can be used to infer new explicit knowledge from implicit knowledge that exists in the domain definition.

When an ontology is produced for a standard such as ebBP, it allows the architects to write expressions based on clear, unambiguous terms and categories. Similar to web documents, which are defined using standard HTML, ontologies also need a standard means of description. The Web Ontology Language (OWL) is a W3C recommendation standard that can be used for expressing ontologies which can be processed by software or used to write or generate it. OWL DL is a sub language of OWL, based on Description Logics and supports those users who need maximum expressiveness while retaining computational completeness which makes it ideal for the ebBP ontology.

Another benefit of using an ontology to support specifications relating to ebBP and its uses is that, once the ontology has been produced, an expression based on that ontology, such as one written using some queries, can be evaluated. One way to evaluate the expressiveness of an ontology is to sketch a set of questions that the ontology must be able to answer. These questions are called competency questions and are considered to be an acceptable means of evaluating expressiveness of an ontology [7], [25].

There are different ways to query an ontology, most popular of which are SPARQL [22] and Protégé OWL DL Query. SPARQL is a query language mainly designed for RDF, which is less expressive than OWL: SPARQL, therefore is not considered as the most appropriate means to query OWL ontologies. Protégé 4.0 (and later versions) provides a DL Query tab which is a powerful and easy-to-use feature for searching a classified OWL DL ontology. The Protégé DL Query language is basically an OWL class expression and is based on the Manchester OWL syntax, a user-friendly syntax for OWL DL.

This approach has benefits where a specification can include such DL queries but also allows a knowledgebase to store and retrieve information related to a process defined using ebBP. It may be one step towards storing

process definitions not just defined with ebBP but also with other business process languages such as BPEL.

This would potentially enhance interoperability and also conformance testing. It would also facilitate long term use of processes defined using ebBP which might continue over decades such that with necessary updates of the ontology it would be possible to make sense of the process definitions even when other technologies have prominence.

IV. RELATED WORK

Business Process interoperability is repeatedly mentioned as one of the most important aspects of B2B integration in the literature. Legner and Wende [11] stress the importance of public process integration in the future success of businesses and suggest that Inter-organisational business process design has to provide concepts to support organisations in aligning the semantics that underlie business processes. They also suggest that compliance with B2B process standards will become more important in the near future. They identify exchangeability of business process models and semantic alignment of business processes as important research agenda items in the future which need to be addressed for more effective and flexible B2B Interoperation.

Gong, Li, Ning, Chen and O'Sullivan [5] introduce inter-organisational business process collaboration as one of the most significant factors in today's global business and recognise the semantic web technologies as a promising direction for integration and collaboration. They provide a semantic agent based approach for achieving inter-organisational process interoperability. Wu and Yang [24] also highlight the importance of ontologies for business process in today's B2B interactions and provide an e-business process modelling framework that outlines the required building blocks for enabling e-business process automation.

A number of works related to ontology development for business processes and workflow languages can also be found in the literature. Examples are oXPDL, an ontology for XPDL [8], an ontology for WS-BPEL [13], an ontology for EPC (of event-driven process chains) [21], an ontology for PetriNet [4] and Business Management Ontology (BMO version 1.0), which is mainly focused on private processes [10].

On the other hand there exist a few ontology related efforts regarding ebXML related specifications. EbXML Registry Profile for OWL [15] provides specifications for publishing and discovering OWL ontologies in the ebXML Registry/Repository. OntologUBL provides an ontology for Universal Business Language [20]. OASIS SET (Support for Electronic Business Document Interoperability Technical Committee) Technical Committee [16] also provides an ontology for the Business Documents, which are based on the ebXML Core Components Technical Specification (CCTS).

Furthermore there are some projects working on semantic aspects of Business Process Management, such as SUPER [1], STASIS [21], and m3po [9]. Table 1 summarises the existing work related to the domain under study.

TABLE 1. A SUMMARY OF RELATED WORK IN THE BUSINESS PROCESS AND ONTOLOGIES AREA.

Standard	Target	Type
WS-BPEL	Private	Process
XPDL	Private	Workflow
EPC	Private	Process
PetriNet	Private	Process
BMO	Private	Process
ebReg/Rep	Public	Registry/Repository
OASIS SET	Public	Business Documents
UBL Ontology	Public	Business Documents - Patterns
Rhizomik ebBP	Public	Business Process
BPMN	Both	Notation Language

None of the works above, except Rhizomik, are targeted at the public aspect of business processes and do not provide a comprehensive ontology for B2B process interoperation. They are either focused on ontologies for private processes or business document and registry aspects of B2B transactions.

The Rhizomik project [3], however, provides facilities for automatic transformation of XML schema and XML documents to RDF and OWL documents respectively. They have specifically mapped an ebBP schema to an OWL ontology. However, with the first examination of the ontology, it is quite clear that it does not cover both the semantics and the syntax of the model. For example, none of the data properties in the ontology have domain and range, none of the Object Properties have domain and most of the Object Properties do not have range. The data types that exist in OWL such as *int*, *string* and *IDREF*, are ignored in this ontology, for each data type a class is defined. This is as a result of automatic translation, without paying attention to the semantics of the entities. Furthermore the way the classes and properties are defined is different from the ebBP ontology presented in this paper, which pays more attention to the semantics. The Rhizomik ebBP ontology therefore, is not able to model a B2B Process in an appropriate way and also is unable to answer the competency questions defined in this paper.

In the next section an ebBP ontology, which is specifically designed for inter-organisational B2B processes, is presented.

V. ONTOLOGY-BASED REPRESENTATION OF EBBP

This section demonstrates the ontology for ebXML Business Process Specification Schema v2.0.4 and its development process. The ebBP ontology is defined using OWL DL ontology language and covers both syntax included in ebBP XML schema and the informal semantics of the ebBP specification. Protégé 4.0.1 is used for developing the ontology, queries are written using Protégé DL query and Pellet is used as a reasoning engine.

To design the ontology two approaches are adopted: top-down and bottom-up. As each approach has its advantages and disadvantages, the combination of the two is believed to result in a more stable and coherent ontology.

The top-down approach starts with identifying the most general concepts, organising them into a high-level taxonomy and system of axioms, and proceed to more specific concepts and axioms. This approach, if conducted appropriately, will cover the basic structure of the domain under study. The top down approach in this paper is taken with focus on ebBP specifications and also in collaboration with standards developers and experts in the field of standardisation, with the aim of covering all the structural concepts.

The first step in the top-down approach defines a UML model of the ebBP entities. This UML model provides a top level and conceptual view of the element in the ebBP Schema. The first set of classes and relationships of the ontology are defined based on this UML diagram.

The second step in the top-down approach ensures the ontology covers all the concepts in the XML Schema. The general rule in this step is to define a class for each element and each complex type in the XML schema. However, to make the ontology more meaningful, this rule is not followed for each and every construct. The ebBP schema is specified using both XML elements and complex types. The latter are hidden in an ebBP XML instance and have little or no semantic value and therefore are ignored for the ontological modelling. This will keep the ontology simple and easier to understand, while covering the semantics. The convention is therefore to use the classes represented in the XML instances and not everything in the schema. This led us to the bottom up approach.

The bottom-up approach starts with the definition of the most specific classes based on the instances available in the real world, with subsequent grouping of these classes into more general concepts. This approach is taken based on instances of XML schema in order to refine the ontology developed in the top-down approach. The first step defines individuals for classes in the ontology, where needed, based on the XML instances. This phase led to discovery of new relationships between classes and also modifications to existing ones. This is illustrated using an ‘ordering process’ in Section C.

For the Object and Data Properties in the ontology, the OWL naming convention is followed, and therefore most of the properties are started ‘has’ or ‘is’ and then the name of the property. For example a relationship between a `BusinessTransaction` and its `RequestingBusinessActivity` in ebBP is modelled through an Object Property called `hasRequestingBusinessActivity`, with `BusinessTransaction` as its domain and `RequestingBusinessActivity` as its range. The exception is when ‘has’ or ‘is’ do not make sense semantically, e.g. `BeginsWith` would have been `hasBeginWith` if the convention was followed, which doesn’t make that much sense.

XML attributes are defined using OWL Data Properties. OWL supports most XML types and therefore the range of the Data Properties are generally set based on the type of XML attributes. However, similar to class definitions, some exceptions are considered in defining the Data Properties; There are attributes in the XML schema whose type is *IDREF*. Following the general rule, they

should be translated to Data Properties with range *IDREF*. *IDREF* is used in XML to refer to an *ID* type defined for another element. However, in the ontology design it doesn’t make sense as we can simply define the range of an Object Property to be another class. For example in the ebBP XML schema, a `DocumentEnvelope` refers to a `BusinessDocument` with `BusinessDocumentRef` attribute which is of type *IDREF*. This should basically match the `nameID` of a `BusinessDocument`, which is of type *ID*. In the ontology however, the `hasBusinessDocument` property of a `DocumentEnvelope` is not defined as a Data Property of type *IDREF*, but as an Object Property with the range `BusinessDocument`. This makes reasoning over the ontology much more precise and makes more sense as the two classes have a proper relationship in the ontology rather than being related based on string matching. In addition the ‘Ref’ part of the property is ignored since it is referring to another class and therefore is not necessary. This also simplifies the ontology.

Running competency questions over the ontology was the final stage of the ontology development. This step also resulted in discovering new relationships and therefore may be considered as a part of the bottom-up approach. An important outcome of this step was to detect structural relationships which do not exist in the XML schema or are not clear enough, e.g. inheritance and inverse properties. This is illustrated in more detail in Section C.

It is important to note that the ebBP ontology is fundamentally different from automatic transformation of an XML schema into OWL. The automatic transformation cannot cover the semantics embedded in both the schema and the textual specifications and leaves some elements without any semantic value. Furthermore it cannot cover the relationships between classes.

Fig. 1 depicts a part of ebBP ontology in three different layouts: Class Definition, Object Properties and Data Properties.

Listing 1 demonstrates two Object Properties of the ontology which depict relationships between different classes of the ontology using OWL Manchester syntax.

A. Competency questions

In order to evaluate the correctness of the ebBP ontology, in this section a set of competency questions are considered to be important to answer. A subset of these competency questions are provided in this section, which are answered in Section C, based on an example process introduced in Section B.

In ebBP, a Business Process is realised by one or more Business Collaborations. Business Collaborations are composed of Business Transactions, which are expressed as exchange of Business Documents. In B2B interactions it is usually very important to know:

1. Which Business Documents are used in a particular Process Specification?
2. Which Business Documents are used in a particular Package?

Class Definitions	Object Properties	Data Properties
<ul style="list-style-type: none"> Thing <ul style="list-style-type: none"> Attachment Attribute Substitution Binary Collaboration Business Transaction Pattern <ul style="list-style-type: none"> Commercial Transaction Business Action <ul style="list-style-type: none"> Requesting Business Activity Responding Business Activity Business Activity Type <ul style="list-style-type: none"> Business Transaction Activity Collaboration Activity Business Collaboration Business Document Business Partner Role Business State Link Type <ul style="list-style-type: none"> From Link To Link Business Transaction <ul style="list-style-type: none"> Collaboration Group Complex Business Transaction Activity Condition Expression Decision Document Envelope Documentation External Roles Failure Fork Join Message Map Multi Party Collaboration Operation Mapping OptName Package Performs Process Specification Role <ul style="list-style-type: none"> Requesting Role Responding Role Signal <ul style="list-style-type: none"> Requesting Role Responding Role Signal Envelope Type <ul style="list-style-type: none"> Acceptance Acknowledgement Acceptance Acknowledgement Exception Receipt Acknowledgement Receipt Acknowledgement Exception Signal Map Specification Start Status Visibility Success Time To Perform Transition 	<ul style="list-style-type: none"> Begins When Ends When Refers To Business Transaction has Attachment has Attribute Substitution has Binary Collaboration has Business Action <ul style="list-style-type: none"> has Requesting Business Activity has Responding Business Activity has Business Activity Type has Business Collaboration has Business Document has Business Partner Role has Business Transaction has Business Transaction Activity has Collaboration Activity has Complex Business Transaction Activity has Condition Expression has Current Role has Decision has Document Envelope has Documentation has External Roles has Failure has From Business State has Multi Party Collaboration has Operation Mapping has Package has Performs has Performs Role has Post Condition has Pre Condition has Requesting Role has Responding Role has Role has Signal has Signal Envelope Type has Specification has Start has Success has To Business State has Variable is Business Action Of <ul style="list-style-type: none"> is Requesting Business Activity Of is Responding Business Activity Of is Business Document Of is Business Transaction Activity Of is Business Transaction Of is Document Envelope Of is Performs Of is Realised By Business Transaction Activity is Role Of is Signal Envelope Type Of <ul style="list-style-type: none"> is Acceptance Acknowledgement Exception Of 	<ul style="list-style-type: none"> has Attribute Name has Binary Collaboration Type Pattern has Business Collaboration Pattern has Business Transaction Head Pattern has Business Transaction BaseType Pattern has Condition Guard has Current Role Ref has DefaultValue has Documentation Lang has Duration has Expression has Expression Language has Fork Type has Instance Version has Interface Name has Legal Intent has Location has MimeType has Name has NameID has Operation Name has Operation Step has Performs Role Ref has Role Ref has Specification Version has Target Namespace has Time To Perform has Type has URI has UUID has Value has pattern is Authenticated is Authorization Required is Concurrent is Confidential is Guaranteed Delivery Required is Inner Collaboration is Intelligible Check Required is Non Repudiation Receipt Required is Non Repudiation Required is Positive Response is Tamper Detectable wait For All

Figure 1. Part of ebBP ontology in three different layouts, Class Definitions, Object Properties and Data Properties.

```

ObjectProperty:
hasRequestingBusinessActivity

Characteristics:
  Transitive

Domain:
  BusinessTransaction

Range:
  RequestingBusinessActivity

InverseOf:
  isRequestingBusinessActivityOf

SubPropertyOf:
  hasBusinessAction

=====
ObjectProperty:
hasAcceptanceAcknowledgement

Characteristics:
  Transitive

Domain:
  BusinessAction
  or RequestingBusinessActivity
  or RespondingBusinessActivity

Range:
  AcceptanceAcknowledgement

InverseOf:
  isAcceptanceAcknowledgementOf

SubPropertyOf:
  hasSignalEnvelopeType

```

Listing 1. Two examples of Object Properties in the ebBP ontology in Manchester Syntax

3. Which collaborations in a particular Process Specification use a Business Document with a specific target namespace?

A Business Transaction in ebBP consists of a Requesting Business Activity, a Responding Business Activity, and one or two Document Flows between partners. A Business Transaction may also involve the exchange of one or more Business Signals that govern the use and meaning of acknowledgements [14]. Fig. 2 depicts the semantics of ebBP Business Transactions.

With regards to fig. 2 the following questions are important to answer:

4. Which signals do the transactions in a particular Business Collaboration use?
5. In which transactions in a particular process does a particular party take a requesting role?

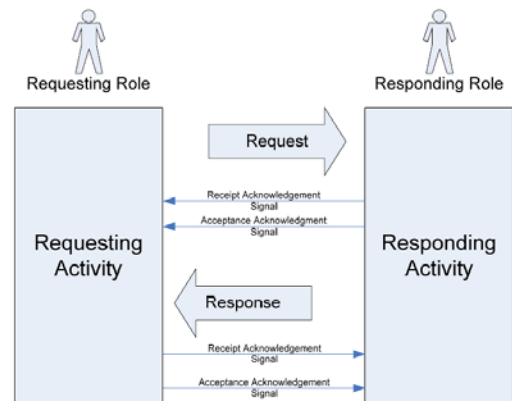


Figure 2. Schematic of core Business Transaction semantics, adopted from [14]

In order to answer these questions for the ebBP ontology an example B2B process is introduced in the following sections and these questions are answered in the context of that process.

B. Motivating example

A Simple Ordering Process is illustrated in fig. 3 which is defined in ebBP v2.0.4 and is based on UBL (Universal Business Language) [17]. This process is publicly available on the OASIS UBL web site at http://docs.oasis-open.org/ubl/cs-UBL-1.0-SBS-1.0/universal-business-process-1.0-ebBP/ebxmlbp-2.0_ubl-1-order-with-simple-response-1.xml. UBL is a library of standard electronic XML business documents such as purchase orders and invoices which is developed by OASIS. This example is used through the rest of this paper to present the ebBP ontology and relevant instances and also to evaluate the ontology with regard to the competency questions. The complete XML instance for the ebBP Ordering process can be found in Appendix 1.

With regards to the ontological representation of ebBP, one should differentiate between the representation of the business process modelling language and the representation of a specific process model. Business process modelling language constructs in an ontology can be represented by classes and properties of an ontology, while specific process models are defined as instances of an ontology. In the ebBP ontology, the language constructs are modelled using OWL and the ordering process itself and its instances are modelled as individuals of the ebBP ontology. To test the expressiveness and correctness of the ontology the above competency questions are answered for this specific process using DL Queries over the ebBP ontology and its individuals.

The ordering Business Process is defined as individuals of the relevant classes and their relationships in the ebBP ontology. Listing 2 shows two individuals of the ontology and depicts their relationship; an instance of a

‘RequestingBusinessActivity’ and an instance of a ‘CommercialTransaction’.

```

Individual: SendOrder_ReqBA

Types:
  RequestingBusinessActivity,
  owl:Thing

Facts:
  hasAcceptanceAcknowledgement AA2,
  hasDocumentEnvelope Order_DE,
  hasReceiptAcknowledgementException
RAE2,
  hasReceiptAcknowledgement RA2,
  hasAcceptanceAcknowledgementException
AAE2,
  isIntelligibleCheckRequired
"true"^^xsd:boolean,
  hasName "Send Order",
  isAuthorizationRequired
"true"^^xsd:boolean,
  isNonRepudiationReceiptRequired
"true"^^xsd:boolean,
  hasNameID "SendOrder_ReqBA",
  isNonRepudiationRequired
"true"^^xsd:Boolean
=====
Individual: CreateOrder_CT

Types:
  CommercialTransaction,
  owl:Thing

Facts:
  hasRequestingRole OrderInitiator,
  hasRequestingRole OrderResponder,
  hasRespondingBsinessActivity
FirmOrder_ResBA,
  hasRequestingBusinessActivity
SendOrder_ReqBA,
  hasName "Create Order",
  hasNameID "CreateOrder_CT"
  
```

Listing 2. Two individuals of ordering process in the ebBP Ontology

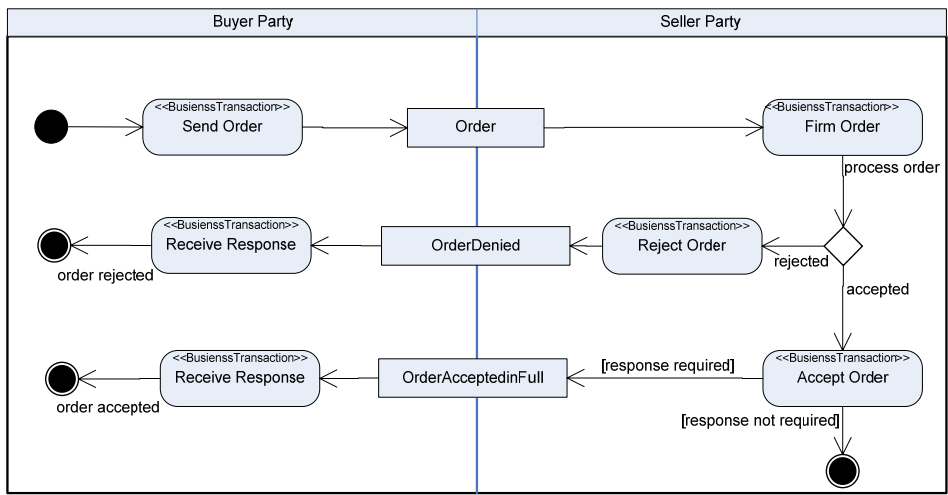


Figure 3. Simple Ordering Process (UBL)

C. Competency questions in the context of ordering process

With regards to the Simple Ordering Process, there are several key 'drill-down' type knowledge questions which are important to answer. In this section the competency questions provided in section B are answered in the context of the Simple Ordering Process. Each competency question is answered using a DL Query provided in the following.

Competency question 1. Which Business Documents are used in the Simple Ordering Process?

```
BusinessDocument and isBusinessDocumentOf
some
  (ProcessSpecification and hasNameUuid value
  "bpid:urn:oasis:names:draft:bpss:ubl-2-order-
  with-simple-response-process-2" )
```

Listing 3. DL Query for competency question 1

The result of this query should be and is: "orderAcceptanceFull_BD", "Order_BD" and "OrderDenied_BD".

As seen in listing 3 the 'isBusinessDocumentOf' Object Property is used for querying the ontology to answer competency question. This property is the inverse property of 'hasBusinessDocument'. Without having this inverse property answering this question would not be possible when there is more than one process defined in the knowledge base. This competency question shows how important inverse properties are for drill down queries. Inverse properties are used in most of the competency questions discussed in this paper.

Competency question 2. Which Business Documents are used in Package "OrderWithSimpleResponse"?

```
BusinessDocument and isBusinessDocumentOf some
  (Package and hasNameID value
  "OrderWithSimpleResponse" )
```

Listing 4. DL Query for competency question 2

The result of this query should be and is: "orderAcceptanceFull_BD", "Order_BD" and "OrderDenied_BD".

Competency question 3. Which Signals do the transactions in collaboration "Create Order" use?

```
Signal and isSignalOf some
  (SignalEnvelopeType and isSignalEnvelopeTypeOf
  some
    (BusinessAction and isBusinessActionOf some
    (CommercialTransaction and
    isBusinessTransactionOf some
    (BusinessTransactionActivity and
    isBusinessTransactionActivityOf some
    (BusinessCollaboration and hasNameID value
    "CreateOrder_BC")))))
```

Listing 5. DL Query for competency question 3

The result of this ontology should be and is: "ra2", "aa2", "aae2" and "rae2".

Competency question 4. Which collaborations in the Simple Ordering process use the Business Document whose target namespace is "urn:oasis:names:specification:ubl:schema:xsd:OrderResponseSimple-2"?

```
(BusinessCollaboration and
  (hasBusinessTransactionActivity some
  (BusinessTransactionActivity and
  refersToBusinessTransaction some
  (BusinessTransaction and hasBusinessAction
  some (BusinessAction and hasDocumentEnvelope
  some (DocumentEnvelope and hasBusinessDocument
  some (BusinessDocument and hasSpecification
  some (Specification and hasTargetNamespace
  value
  "urn:oasis:names:specification:ubl:schema:xsd:O
  rderResponseSimple-2" ^^ anyURI ))))))))
and
(BusinessCollaboration and
isRealisationOfProcessSpecification some
  (ProcessSpecification and hasUuid value
  "bpid:urn:oasis:names:draft:bpss:ubl-2-order-
  with-simple-response-process-2" ))
```

Listing 6. Optimised DL Query for competency question 4

The result of this query should be and is "CreateOrder_BC".

Object Property hasBusinessAction is defined as a Super property of hasRequestingBusinessActivity and hasRespondingBusinessActivity in the ebBP Ontology. Furthermore they all have inverse properties called isBusinessActionOf, isRequestingBusinessActivityOf and isRespondingBusinessActivityOf respectively. This allows the competency questions to be answered. Additionally if the super property didn't exist the query in listing 6 would have been as listing 8.

It is clear that although it would have been possible to answer the competency question 4 without optimisation, the query would have been longer and less clear. This was achieved with a simple super property added to the ontology. These added semantics are only achievable through systematic engineering of the

ontology and impossible with automatic transformation of a XML schema to an ontology.

Competency question 5. In which transactions in the Simple Ordering process does the "Buyer" party take a requesting role?

```
CommercialTransaction and
  isBusinessTransactionOf some
    (ProcessSpecification and hasUuid value
"bpid:urn:oasis:names:draft:bpss:ubl-2-order-
with-simple-response-process-2")
and
  isRealisedByBusinessTransactionActivity some
    (BusinessTransactionActivity
      and hasPerforms some
        (Performs and hasCurrentRole some
          (Role and hasName value "Buyer")
        and hasPerformsRole some
          (RequestingRole and
            hasNameID value "OrderInitiator")))
```

Listing 7. DL Query for competency question 5

```
BusinessCollaboration and
  (hasBusinessTransactionActivity some
    (BusinessTransactionActivity and
      refersToBusinessTransaction some
        (CommercialTransaction and
          (hasRequestingBusinessActivity some
            (RequestingBusinessActivity and
              hasDocumentEnvelope some
                (DocumentEnvelope and
                  hasBusinessDocument some
                    (BusinessDocument and hasSpecification
                      some (Specification and hasTargetNamespace
                        value
"urn:oasis:names:specification:ubl:schema:xsd
:OrderResponseSimple-2" ^^ anyURI ))))))))
or
BusinessCollaboration and
  (hasBusinessTransactionActivity some
    (BusinessTransactionActivity and
      refersToBusinessTransaction some
        (CommercialTransaction and
          (hasRespondingBsinessActivity some
            (RespondingBusinessActivity and
              hasDocumentEnvelope some
                (DocumentEnvelope and
                  hasBusinessDocument some
                    (BusinessDocument and hasSpecification
                      some (Specification and hasTargetNamespace
                        value
"urn:oasis:names:specification:ubl:schema:xsd
:OrderResponseSimple-2" ^^ anyURI ))))))))
and
BusinessCollaboration and
  isRealisationOfProcessSpecification some
    (ProcessSpecification and hasUuid value
"bpid:urn:oasis:names:draft:bpss:ubl-2-order-
with-simple-response-process-2")
```

Listing 8. Not Optimised DL Query for competency question 4

Competency question 5 is basically addressing the relation between Business Transactions and Business Transaction Activities, which are their realisation and their corresponding roles.

The result of this query should be and is CreateOrder_CT.

VI. CONCLUSION AND FUTURE RESEARCH

This paper presents an ontology for ebXML Business Process Specification Schema (ebBP) which is a public B2B process standard, developed by OASIS. This ontology is much richer than an automatic transformation of an XML schema to OWL and captures syntactic and semantic aspects of ebBP, extracted from ebBP XML Schema as well as informal specifications. The ebBP ontology is intended to facilitate standards-based B2B interoperability and is evaluated against a set of competency questions. This can greatly facilitate B2B process alignment between trading partners.

Next steps in this research are as follows:

- Add rules to the ebBP Ontology, where applicable.
- Explore the possibility of add test assertions to the ontology in order to conduct interoperability testing.
- Developing an upper ontology for B2B processes, which covers ebBP processes and is also general enough to cover all B2B transactions.
- Explore how ontologies can be utilised in the process of developing B2B process standards.
- Conduct a more coherent evaluation of the ontology based on industrial data.
- Explore how ebBP ontology can be integrated with Business Process patterns standards such as UBL and UBP.
- Explore how B2B processes can become context aware.

VII. ACKNOWLEDGEMENT

The work published in this paper is sponsored in part by Fujitsu Laboratories of Europe.

VIII. REFERENCES

- [1] SUPER (Semantics Utilised for Process Management within and between Enterprises) Integrated Project, 2009.
- [2] Firat, A., Madnick, S., Manola, F., Multi-Dimensional Ontology Views Via Contexts in the ECOIN Semantic Interoperability Framework. AAAI Workshop - Technical Report, 2005, PP. 1-8.
- [3] Garcia, R., & Gil, R., "Facilitating Business Interoperability from the Semantic Web", Lecture Notes in Computer Science, 4439 LNCS, 2007, PP. 220-232.
- [4] Gašević, D., & Devedžić, V., Petri Net Ontology. Knowledge-Based Syst., 2006, PP. 220-234.
- [5] Gong, R., Li, Q., Ning, K. et al., "Business Process Collaboration using Semantic Interoperability: Review and Framework", Lecture Notes in Computer Science, 4185 LNCS, 2006, PP. 191-204.
- [6] Gruber, T. R., "Toward Principles for the Design of Ontologies used for Knowledge Sharing?", International Journal of Human - Computer Studies, 43, 1995, PP. 907-928.

- [7] Gruninger, M., & Fox, M. S., "Methodology for the Design and Evaluation of Ontologies", Workshop on Basic Ontological Issues in Knowledge Sharing, 1995.
- [8] Haller, A., Gaaloul, W., Marmolowski, M. "Towards an XPDL Compliant Process Ontology", Proc.IEEE Congress on Services, SERVICES 2008, PART 1, PP. 83-86.
- [9] Haller, A., Oren, E., Kotinurmi, P., "An Ontology for Internal and External Business Processes", Proc. of the 15th International Conference on World Wide Web, 2006, PP. 1055-1056.
- [10] Jenz, D. E., Ontology-Based Business Process Management. Strategic White Paper - Draft, Erlensee, 2003.
- [11] Legner, C., & Wende, K., "The Challenges of Inter-Organizational Business Process Design - A Research Agenda.", Proc. 15th European Conference on Information Systems, 2007 PP.106-118.
- [12] Milner, R., "A calculus of communicating systems.", Springer Verlag 1980.
- [13] Nitzsche, J., Wutke, D., Van Lessen, T., "An Ontology for Executable Business Processes.", Workshop SBPM, 2007, PP. 52-63.
- [14] OASIS EbXML Business Process Specification Schema Technical Specification (Version 2.0.4), 2006.
- [15] OASIS ebXML Registry Profile for Web Ontology Language (OWL) Version 1.5., 2006.
- [16] OASIS Semantic Support for Electronic Business Document Interoperability (SET), 2009.
- [17] OASIS Universal Business Language (UBL) v2.0., 2008.
- [18] Rebstock, M., Fengel, J., Paulheim, H. (eds.), "Ontologies-based business integration.", Springer Berlin Heidelberg, 2008.
- [19] Singh R., Iyer L.R., Salam A.F., "The Semantic E-Business Vision." Commun ACM, 2005, PP. 38-41.
- [20] The Ontolog Forum: UBL Ontology, 2003.
- [21] Thomas, F., & Fellmann, M., "Semantic Business Process Management: Ontology-Based Process Modeling using Event-Driven Process Chains.", International Journal of Interoperability in Business Information Systems, 2007, pp. 29-44.
- [22] W3C SPARQL Query Language for RDF, 2008.
- [23] W3C Semantic Web Activity.
- [24] Wu, B., Li, L., Yang, Y., "Ontological Approach Towards E-Business Process Automation." Proc. IEEE International Conference on e-Business Engineering, ICEBE 2006, pp. 154-161.
- [25] Yu, J., Thom, J. A., Tam, A., "Requirements-Oriented Methodology for Evaluating Ontologies." Inf Syst, 2009, pp. 686-711.

IX. APPENDIX I

The following is the UBL Simple Ordering Process which can also be found online at: http://docs.oasis-open.org/ubl/cs-UBL-1.0-SBS-1.0/universal-business-process-1.0-ebBP/ebxmlbp-2.0_ubl-1-order-with-simple-response-1.xml

```
<ProcessSpecification xmlns="http://docs.oasis-open.org/ebxmlbp/ebbp-2.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  name="Order Process"
  nameID="OrderWithSimpleResponseProcess_PS"
  specificationVersion="2" instanceVersion="0.9.7"
  xsi:schemaLocation=" http://docs.oasis-open.org/ebxmlbp/ebbp-2.0"
  uuid="bpid:urn:oasis:names:draft:bpss:ubl-2-order-with-simple-response-process-2">
  <Signal nameID="ra2" name="ReceiptAcknowledgement">
    <Specification nameID="rabps2"
      name="ReceiptAcknowledgement" location="ebbp-signals-2.0.xsd"/>
  </Signal>
  <Signal nameID="rae2"
    name="ReceiptAcknowledgementException">
```

```
    <Specification nameID="raebps2" name="Exception"
      location="ebbp-signals-2.0.xsd"/>
  </Signal>
  <Signal nameID="aa2" name="AcceptanceAcknowledgement">
    <Specification nameID="aabps2"
      name="AcceptanceAcknowledgement" location="ebbp-signals-2.0.xsd" />
  </Signal>
  <Signal nameID="aae2"
    name="AcceptanceAcknowledgementException">
    <Specification nameID="aaebps2" name="Exception"
      location="ebbp-signals-2.0.xsd"/>
  </Signal>
  <Package name="Ordering" nameID="Ordering">
    <BusinessDocument name="Order" nameID="Order_BD">
      <Specification
        targetNamespace="urn:oasis:names:specification:ubl:schema:xsd:Order-2"
        name="Order" nameID="Order_SP"
        location="http://docs.oasis-open.org/ubl/cs-UBL-2.0/xsdrt/maindoc/UBL-Order-2.0.xsd"
        type="schema"/>
      </BusinessDocument>
      <BusinessDocument name="Order Denied"
        nameID="OrderDenied_BD">
        <ConditionExpression expressionLanguage="XPath1"
          expression="//AcceptedIndicator='false'"/>
      </Specification>
      targetNamespace="urn:oasis:names:specification:ubl:schema:xsd:OrderResponseSimple-2"
      name="Order Response Simple - Order Denied"
      nameID="OrderResponseSimpleOrderDenied_SP"
      location="http://docs.oasis-open.org/ubl/cs-UBL-2.0/xsdrt/maindoc/UBL-OrderResponseSimple-2.0.xsd"
      type="schema"/>
      </BusinessDocument>
      <BusinessDocument name="Order Accepted In Full"
        nameID="OrderAcceptedInFull_BD">
        <ConditionExpression expressionLanguage="XPath1"
          expression="//AcceptedIndicator='true'"/>
      </Specification>
      targetNamespace="urn:oasis:names:specification:ubl:schema:xsd:OrderResponseSimple-2"
      name="Order Response Simple - Order Acceptance"
      nameID="OrderResponseSimpleOrderAcceptance_SP"
      location="http://docs.oasis-open.org/ubl/cd-UBL-2.0/xsdrt/maindoc/UBL-OrderResponseSimple-2.0.xsd"
      type="schema"/>
      </BusinessDocument>
      <CommercialTransaction name="Create Order"
        nameID="CreateOrder_CT"
        isGuaranteedDeliveryRequired="true">
        <RequestingRole
          name="InitiatorOrderWithSimpleOrComplexResponse"
          nameID="OrderInitiator"/>
        <RespondingRole
          name="ResponderOrderWithSimpleOrComplexResponse"
          nameID="OrderResponder"/>
        <RequestingBusinessActivity name="Send Order"
          nameID="SendOrder_ReqBA"
          isAuthorizationRequired="true"
          isIntelligibleCheckRequired="true"
          isNonRepudiationReceiptRequired="true"
          isNonRepudiationRequired="true">
          <DocumentEnvelope name="UBL 2 Order"
            nameID="Order_DE" businessDocumentRef="Order_BD"
            isAuthenticated="transient" isConfidential="transient"
            isTamperDetectable="transient"/>
          <ReceiptAcknowledgement signalDefinitionRef="ra2"
            nameID="RA2" name="RA2"/>
          <ReceiptAcknowledgementException
            signalDefinitionRef="rae2" nameID="RAE2" name="RAE2"/>
```

```

    <AcceptanceAcknowledgement signalDefinitionRef="aa2"
nameID="AA2" name="AA2"/>
    <AcceptanceAcknowledgementException
signalDefinitionRef="aae2" nameID="AAE2" name="AAE2"/>
</RequestingBusinessActivity>
    <RespondingBusinessActivity name="Firm Order"
nameID="FirmOrder_ResBA"
    isAuthorizationRequired="true"
isIntelligibleCheckRequired="true"
    isNonRepudiationReceiptRequired="true"
isNonRepudiationRequired="true">
    <DocumentEnvelope name="Denied"
nameID="OrderDenied_DE"
businessDocumentRef="OrderDenied_BD"
    isPositiveResponse="false" isAuthenticated="transient"
isConfidential="transient"
    isTamperDetectable="transient"/>
    <DocumentEnvelope name="Accepted In Full"
nameID="OrderAcceptedInFull_DE"
    businessDocumentRef="OrderAcceptedInFull_BD"
isPositiveResponse="true"
    isAuthenticated="transient" isConfidential="transient"
isTamperDetectable="transient"/>
    <ReceiptAcknowledgement signalDefinitionRef="ra2"
nameID="RA2_B" name="RA2"/>
    <ReceiptAcknowledgementException
signalDefinitionRef="rae2" nameID="RAE2_B" name="RAE2"/>
    <AcceptanceAcknowledgement signalDefinitionRef="aa2"
nameID="AA2_B" name="AA2"/>
    <AcceptanceAcknowledgementException
signalDefinitionRef="aae2" nameID="AAE2_B" name="AAE2"/>
</RespondingBusinessActivity>
</CommercialTransaction>
    <BusinessCollaboration name="Create Order"
nameID="CreateOrder_BC">
    <Role name="Buyer" nameID="Buyer"/>
    <Role name="Seller" nameID="Seller"/>
    <Start name="Start Send Order"
nameID="CreateOrder_ST">
    <ToLink toBusinessStateRef="CreateOrder_BTA"/>
</Start>
    <BusinessTransactionActivity name="Create Order"
nameID="CreateOrder_BTA"
    businessTransactionRef="CreateOrder_CT"
hasLegalIntent="true">
    <Performs currentRoleRef="Buyer"
performsRoleRef="OrderInitiator"/>
    <Performs currentRoleRef="Seller"
performsRoleRef="OrderResponder"/>
</BusinessTransactionActivity>
    <Decision name="Create Order"
nameID="CreateOrder_Decision">
    <FromLink fromBusinessStateRef="CreateOrder_BTA"/>
    <ToLink toBusinessStateRef="CreateOrder_Failure">
    <ConditionExpression
expressionLanguage="DocumentEnvelope"
expression="OrderDenied_DE"/> </ToLink>
    <ToLink toBusinessStateRef="CreateOrder_Success">
    <ConditionExpression
expressionLanguage="DocumentEnvelope"
expression="OrderAcceptedInFull_DE"/> </ToLink>
</Decision>
    <Failure name="Failure" nameID="CreateOrder_Failure"/>
    <Success name="Success" nameID="CreateOrder_Success"/>
</BusinessCollaboration>
</Package>
</ProcessSpecification>

```