

# Extraction of Topic Map Ontology for Web Service-Oriented Enterprises

Suman Roy<sup>1</sup>(✉), Kiran Prakash Sawant<sup>2</sup>, Aditya Kale<sup>3</sup>,  
and Olivier Maurice Charvin<sup>4</sup>

<sup>1</sup> Infosys Ltd., # 44 Electronics City, Hosur Road, Bangalore, India

Suman.Roy@infosys.com

<sup>2</sup> Department of CSE, IIT Bombay, Mumbai, India

krnswn@cs.cse.iitb.ac.in

<sup>3</sup> Visa Inc Technology Center, Bagmane Tech Park, Bangalore, India

kaleaditya92@gmail.com

<sup>4</sup> Technical University of Munich, Munich, Germany

Olivier.Charvin@tum.de

**Abstract.** A Service Oriented Enterprise (SOE) is a new model of organization linking business processes and IT infrastructure across the enterprise. It can be enabled through the deployment of Service Oriented Architectures (SOA). At the heart of SOA are the services that are orchestrated using message passing, action coordination etc., web services being an example. However, there is almost no standard business semantics of web services which makes them isolated and opaque. In order to provide a common understanding of business of each other organizations are using trading exchange languages like Universal Business Languages (UBL). Although, these standards provide syntactic interoperability, they do not support efficient sharing of conceptualizations. Ontology can play an important part here, by providing a formal approach to specify shared conceptualization, and thus enabling semantic interoperability. This paper presents an approach for ontology modeling for business process standards used in B2B transactions in web services in terms of a semantic web formalism, *viz.* Topic Map.

**Keywords:** Service oriented enterprise (SOE) · Web services · Business process · UBL · XML · Topic Map · Ontology · Tolog

## 1 Introduction

A Service Oriented Enterprise (SOE) is a model of organization whose business processes and IT infrastructure are integrated across the enterprise to

---

Olivier Charvin, then a student of École Polytechnique, PALAISEAU Cedex, France, was an InStep intern with Infosys Labs during July-Aug'12 and Aditya Kale, then a student of dept of CSE at IIT Kharagpur, did his internship with Infosys Labs during May-June'13.

deliver on-demand services to customers, partners and suppliers. It can benefit the organization by showing its agility to respond quickly to constantly changing business scenario. SOE combines internet technologies and business process management in a three layer model: Enterprise Performance Layer, Business Process Management Layer and the underlying Service Oriented Architecture (SOA). SOA consists of a collection of services communicating with each other through transferring simple data or coordinating on the same action. Web services provide the connection technology of SOA in the sense that it is designed to support interoperable machine-to-machine interaction over a network. However, there are commonly known inhibitors of web services adoption, which includes a lack of semantic consistency in business processes such as ordering and billing, and absence of work-flow mechanisms to orchestrate a group of specialized web services in support of a single web service.

In a global business environment of changing business values these days, organizations are finding it beneficial to interact with other trading partners for which they need to agree on a common business semantics in a web services environment. They should understand the business processes of each other and align the processes to the common need of organizations using a common language. Example of trading exchange languages are UBL and ebXML, - they define how enterprises can conduct business across the globe, removing barriers associated with distance and language with e-commerce facilitated by Web services. Universal Business Language (UBL) [15], an OASIS initiative, is designed to standardize common business documents as well as processes, which can facilitate B2B integration. It is sponsored by governments and tested in large-scale deployments supporting cheap, painless e-commerce transactions between enterprises of all sizes. It has a library of reusable components such as Address and Price, and a set of document schema such as Order and Invoice to be used in E-business, using XML. Although UBL acts as a medium of common language between their users, the XML documents reflect only syntax and not the semantics of transactions. Effectively managing the data stored in these artifacts is of utmost importance to any company for not only business accountability, but for information retrieval, discovery and auditing. Therefore, it is necessary to add structure and semantics to provide a mechanism for precisely describing data in UBL business documents. Such semantic information can be defined effectively through ontology languages like, semantic web initiatives [1], OWL, Topic Map [7] etc.

Ontologies are usually rooted in a logic-based formalisms which can capture precise and consistent descriptions of classes, instances, properties and associations, and reason about them. These ontologies include a wide range of models of varying degree of semantic richness and complexity. Topic Map, a form of semantic web technology is a standard for the representation and interchange of knowledge with a suitable query mechanism [7, 12]. While DL-based languages like OWL-DL [1] reside on the upper level of the ontology spectrum, formalisms like RDF and Topic Map (TM) are at the middle level of the spectrum offering weaker semantic models. TM is simple to use, requires less storage space, renders easy visualization and facilitates faster query processing. Hence, for most of

the business standards used in B2B exchange through Internet, we believe that Topic Map would be the appropriate formalism for ontology extraction.

We use a top-down approach for ontology design which starts with identifying the most general concepts, organizing them into a high-level taxonomy and system of axioms and proceed to more specific concepts and axioms. We first define a Topic Map model of entities in UBL processes, - the basic set of topics and associations of the ontology are defined, which is followed by constructing a Topic Map ontology for capturing flow diagram of UBL processes. In the next step, we ensure all the concepts in the UBL document schema are captured in the ontology. Finally all the Topic Maps are merged using common concepts/topics.

The paper is organized as follows. We shall quickly dispense with preliminaries in Sect. 2. While we provide a mapping of UBL processes to Topic Map in Sect. 3 we briefly discuss the translation of UBL schema to Topic Map ontology in Sect. 4. We undertake a performance evaluation of our approach in Sect. 5. A case study on Freight Billing process is described in Sect. 6. Finally we conclude in Sect. 7.

*Related work.* Semantic interoperability in business processes is one of the major themes in B2B integration in web services literature. Lenders and Wende [10] have suggested that inter-organizational business process design should be driven more by semantics. Gong *et al.* [3] identify semantic web technologies as important vehicle for integration and collaboration of business processes. Moreover, a semantic agent-based approach is proposed by them for achieving cross-organizational interoperability. Wu and Yang [16] mention the importance of ontology in business process design and provide a modeling framework for E-business in terms of building blocks that aid process automation.

There are works related to ontology development for business processes. Examples include ontology for WS-BPEL [11], OWL-DL ontology for business processes [4], OntologyUBL [13] to design an ontology for UBL, to name a few. In [6], Heravi, Bell and Lycett propose an ontology for ebBP schema with a view to capture semantics embedded within B2B processes, thus enabling reasoning over the shared concepts. However, none of the works provides a comprehensive ontology for B2B process interactions.

In [17] Yarimagan et al. discuss a method to enrich UBL with semantics-based translations for maintaining interoperability between schema documents. They propose a component ontology for UBL using OWL for representing the semantics of individual components and their relationship within schema. However, their ontology scheme is not easy to understand and not amenable to visualization. In this work, we try to alleviate these problems by using Topic Map for ontology modeling of UBL, which offers advantages like reduced storage space, less conversion time, easy visualization and simple querying. Topic Map has been earlier used for extracting knowledge out of UBL documents which has been later used for information retrieval purposes [8]. This work aims to sharpen and generalize the conversion technique defined in that work.

## 2 Preliminaries

**Universal Business Language (UBL).** In this paper, we are using one particular business specification language called Universal Business Language (UBL), which is an OASIS standard providing a library of reusable component schema defined using XML Schema. UBL comprises of business processes (activity diagrams with swim-lanes/roles) and embedded schema documents, containing a set of information components that are exchanged as part of a business transaction such as placing an order etc.

UBL is being used by several communities around the globe [17]. As UBL is being adapted in different industries, across geopolitical and regulatory contexts, it will be useful to customize UBL schema as per the need of user community. However, these customizations may be a hindrance to interoperability as communities may prefer using non-standard schema. With a view to providing a solution to this problem we propose a method to extract ontology out of UBL that can provide efficient semantic interoperability. This ontology extraction works in two parts, UBL processes are first converted to Topic Map ontology and then UBL schema are translated to Topic Map ontology. Finally these Topic Maps are merged to generate the final ontology. When two parties from different communities want to make a business transaction by using the artifacts tailored for their purposes they can still use them, and still maintain the interoperability by using the generated ontologies from these artifacts.

**Topic Map.** Topic Map (TM) is an ISO standard for knowledge representation and interchange with an emphasis on navigation and retrieval of information. As defined by the standard, Topic Map can be used to represent information as topics. It links topics together in a meaningful way to facilitate navigation and filtering of information [7]. In general, a Topic Map portrays groupings of addressable information objects around topics (“occurrences”), and relationships between topics (“associations”). A topic has three characteristics; names, occurrences and roles in associations. A topic can possess one or more names. A topic is linked to related information resources through occurrences. Moreover, a topic can be associated with other topics through roles in associations. While a topic can be an instance of zero, one, or, more classes, these classes themselves correspond to topics. An occurrence can be captured as an instance of one class, but not necessarily. Similarly, for an association. Some topics play a role in an association, they are called role playing topics or short role players.

## 3 Translation of UBL Process to Topic Map

UBL processes are captured using mostly the constructs of standard UML Activity Diagrams. With this one can associate a Business Process Diagram (BPD) which is based on flowchart related ideas, and provides a graphical notation for business process modeling. In a flow graph the control flow relation linking two nodes is represented by a *directed edge* capturing the execution order between tasks of a BPD. A *node* can be a task (also called an activity), an event or a

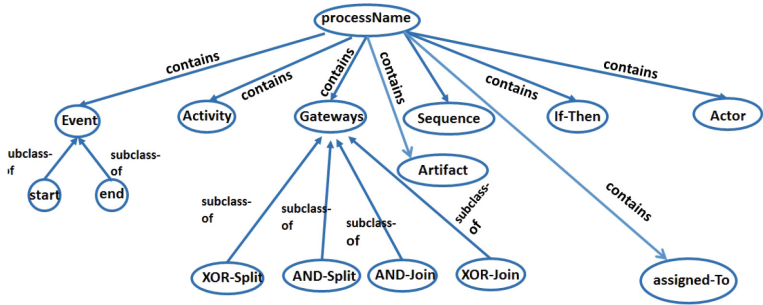
split/join gateway. In a BPD, there is a *start event* denoting the beginning of a process, and *end events* denoting the end of a process. A sequence is made of a collection of nodes each of which has an incoming and an outgoing arc. The gateway (control node) is represented by a diamond with a ‘+’ sign inside for an AND-gateway, and a ‘×’ sign inside for an XOR-gateway. A *fork* (AND-split) and a *synchronizer* (AND-join) portray their usual meaning. Similarly for, *choice* (XOR-split) and a *merge* (XOR-join). One can define swim-lanes/partitions for these processes just like partitions for activity diagrams; swim-lanes depict those actors/agents that are responsible for execution of particular tasks.

We shall now impose a few syntactic restrictions on process models to reduce ambiguity and lack of well-formedness. A UBL process (or a process) is *well-formed* (like a well-formed business process in [5]) if and only if there is only one outgoing edge from a start event, and only one incoming edge to an end event, there is only one incoming edge to a task and exactly one outgoing edge out of a task, every fork and choice has exactly one incoming edge and at least two outgoing edges, every synchronizer and merge has at least two incoming edges and exactly one outgoing edge, and every node is on a path from a start node to some end node.

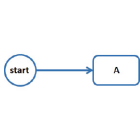
A pattern-oriented method of process modeling and retrieval was proposed in [4] using OWL-DL representation. We adopt a similar approach for ontology creation using Topic Map, wherein we decompose the process in several patterns and generate Topic Map descriptions for each of them. Then using the connectivity information of the processes we tie the Topic Map for the patterns together and generate the whole Topic Map. We also maintain a Topic Map model which describes activities, events, gateways and their hierarchies. In the end we merge these two Topic Maps to get the overall Topic Map for an individual process. We admit that the presented mapping is a syntactic one, however the semantics of control flow of business process can be captured rigorous formalisms (with higher querying time) [4] which is beyond the scope of this work.

We consider basic control flow patterns (as suggested in [14]) of business processes and translate them to corresponding Topic Map preserving control flow of the process diagram. We create some basic topics corresponding to events, actors, artifacts, activities and gateways which are connected to a main topic called “processName” through *contains* association. There are only two kinds of events: a unique Start event and End events. A gateway will contain Split and Merge gateways. Also a Split gateway will contain XOR-split and AND-split gateways. Same holds for Merge gateways too. In cases of process having swimlanes, Actors are modeled as topics contained in the Topic Map model. Moreover, an Activity (Task) can be connected to an Actor through *assignedTo* association. A Topic Map Model for basic process elements is shown in Fig. 1(a).

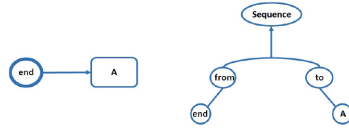
A sequence of two nodes is modeled as an association class called *sequence* which connects association roles *from* and *to*. Depending on the direction of the connectivity of two nodes, they are suitably connected to association roles ‘from’ and ‘to’, see Fig. 1(d). A Start event and an End event with succeeding and preceding nodes respectively are modeled in a similar fashion, see Fig. 1(b) and 1(c). An AND-split is modeled as an association class *AND-split* linked to



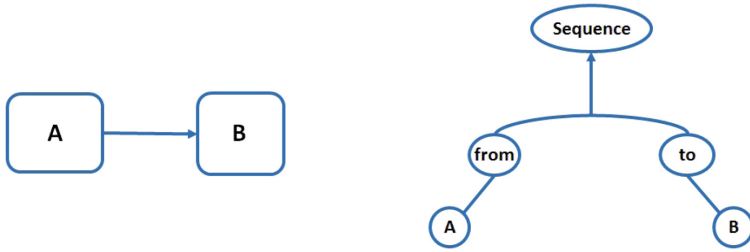
(a) TM for basic process elements



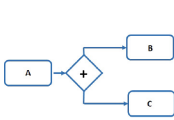
(b) TM for a start event



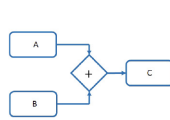
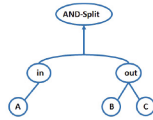
(c) TM for an end event



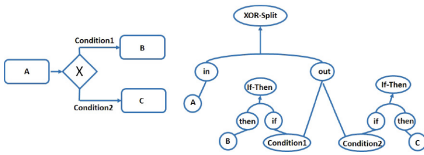
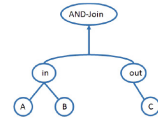
(d) TM for a sequence node



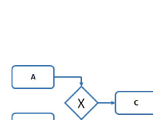
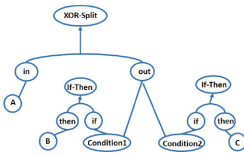
(e) TM for a an AND-split



(f) TM for an AND-join



(g) TM for a XOR-split



(h) TM for a XOR-join

**Fig. 1.** Mapping of process patterns to Topic Map

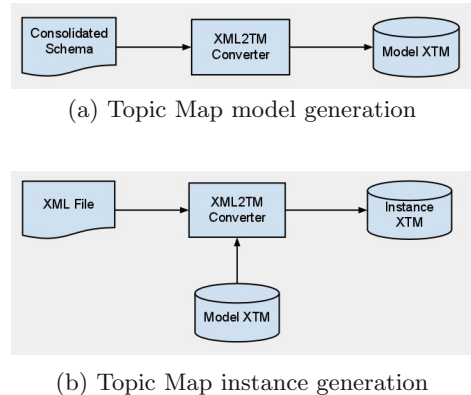
two association roles *in* and *out*, see Fig. 1(e). Similarly, an AND-join is modeled as an association class *AND-join* linked to two association roles *in* and *out*. The role ‘in’ will be connected to incoming node topics and the ‘out’ will be connected to one outgoing topic node, see Fig. 1(f). An XOR-split is captured like an AND-split, however the association role ‘out’ is connected to Condition topics. One association class *if-then* is connected to roles *if* and *then*. The role ‘if’ is connected to condition topic coming out of the split association role and ‘then’ association role is connected to the appropriate topic node arising out of the occurrence of this condition, see Fig. 1(g). A XOR-join is modeled similar to an AND-join, see Fig. 1(h).

## 4 Translation of UBL Schema to Topic Map Ontology

UBL makes use of a set of XML Schema documents to define various types of XML files that are used for B2B purposes. All allowed B2B communications take place via exchange of XML files adhering to these schema files.

**Translation of XML to Topic Map.** XML resembles a tree model, where in nodes are labeled and outgoing edges are ordered. However, Topic Map is based on graph model, where almost everything is modeled as a topic (node) which are linked with association, with other topics playing association roles. Our mapping exploits the relational structure of the XML schema and turns a link between nodes into an association. We assume that an XML schema (XSD) is always available with each XML document. The conversion can be divided into two tasks, TM model extraction from XML Schema and instance generation from one or more XML instances (see Fig. 2).

In many cases XML schema contains “import” or “include” constructs. UBL documents [15] have multi-layered transitive imports. Also, internal references are given by the construct *ref* in XSD files. These references are common features in XSDs and handling them could be problematic for any XSD to ontology converter. Our converter is able to handle internal references in XML Schemas, both in element and type. As a pre-processing step we remove import constructs and internal references using *schema consolidation* described in detail in [9].



**Fig. 2.** Architecture of XML to Topic Map converter

**XSD to Topic Map Model.** Each XML schema `complexType` is mapped to an association topic. For example, compositors like `sequence`, `choice` and `all` are mapped to the respective association classes like `sequence`, `choice` and `all`.

The root member of this compositor is mapped to the association role ‘has’ which is linked with the corresponding role playing topic. For “sequence” type, other members are mapped to association roles `member(1)`, `member(2)`, ... in the proper order. In the case of type “all” each member is mapped to the association role `member`, while for type “choice” all the members are mapped to the association role `altmember`. Again all the association roles are properly connected to role playing topics. A role playing topic corresponding to `has` role will be linked with a topic via an association `instanceof` later. The other role playing topics are connected to occurrence topics suitably. For a `simpleType` compositor, the main element is connected to occurrence types which takes a particular occurrence value. An `attribute` or element `declaration` is mapped to suitable occurrences. Some of these occurrences may be an instance of a class. Model group definitions and attribute group definitions are specializations of complex types and hence, they are handled in a similar fashion. It may be noted that a type in a XSD can be suitably mapped to a topic type.

Cardinalities are handled in a special way. An element having `minOccurs` or `maxOccurs` with integer type can be mapped to an occurrence of interval type. These occurrences can be seen as a subclass of a topic of the set of all intervals of integers. In case of an element having only `maxOccurs` the mapped interval is bounded below with 0 as the left boundary point. Similarly, for element having only `minOccurs` the mapped interval is bounded above with  $\infty$  as the right boundary point. Table 1 highlights the key aspects of the mapping.

**Table 1.** Mapping between elements of XSD and Topic Map

XSD	Topic Map
xsd:elements, containing other elements or having at least one attribute	tm:Topic Types coupled with tm:Association Types
xsd:elements, with neither sub-elements nor attributes	tm:Occurrence Types
xsd:minOccurs, xsd:maxOccurs	tm:Occurrence Types
xsd:sequence	tm:Association “seq” with member1, . . . ,membern
xsd:all	tm:Association “all” with member association role
xsd:choice	tm:Association “choice” with altmember association role

**XML to Topic Map Instance.** There may exist several XML instances for a XSD file. All such XML instances need to be added to the Topic Map ontology. We only gather information related to actual values from XML files, as all other information regarding the Topic Map model ontology will already be picked from its corresponding XSD file. In order to uniquely identify each XML instance



added to the TM ontology, we assume that the name of the XML instance file will be unique with regards to other such XML instance file names. Based upon this assumption we assign this file name, without the *.xml* extension, as the name of the instance which is to be added to the ontology. Further, this name will be used as a prefix for creation of instances of the topic types contained within the Topic Map model ontology. So if we have *note1.xml* as the XML instance file name, then we create *note1* as the instance to be added to the TM ontology. Also, if there exists a topic type *person*, then we create *note1\_person* as its instance gathered from the corresponding XML file *note1.xml*. This way we make sure that each and every instance file can be uniquely identified in the TM ontology.

The root element of XML file will become the top element of both the Topic Map model and the Topic Map instance, which are connected to XML elements through *contains* association. This will help in merging the model and instance Topic Map to get the final Topic Map Ontology.

**Gaps in the Translation: Future Work.** Currently we are not handling some language components like, `abstract`, `final`, `block`, `default`, `form`, wildcard identity constraints definitions and `complexType`s derived by restriction from `simpleTypes`. Some of these are not commonly used, and others can not be handled using the current feature of Topic Map, which will not be detrimental for further development.

## 5 Performance Measures

In this section we consider performance issues for our translation. We used a laptop with Windows 7 (64-bit) on Intel Core i3-2310M processor with 2.10 GHz speed, 4 GB RAM and 320 GB of hard disk capacity for the evaluation.

We consider the dataset UBL-1.0<sup>1</sup>. The size of Topic Map generated for UBL processes are small in size, and hence we do not consider their performance. In this dataset we have 8 XML schema files, out of which only 7 contain the corresponding XML files. We consider these 7 XSD/XML file combinations for conversion. These schema files contain a very high level of recursive imports and are first subjected to Schema Consolidation and then the consolidated schema is given as input to the converter. The performance measures for the UBL dataset are given in Table 2.

Figure 3(a) shows a graph which compares the time taken for the conversion to XTM (Serialization of Topic Map) Model ontology with the Consolidated Schema size. This graph shows that as the size of the Consolidated Schema increases, the time taken for the Model Generation increases linearly. The size of the generated XTM Model Ontology also increases linearly when compared to the Consolidated Schema size. In Fig. 3(b), we can see that the performance is similarly linear for Topic Map instance ontology generation. We could not compare our result with that of [17] as we could not find an online implementation of that work.

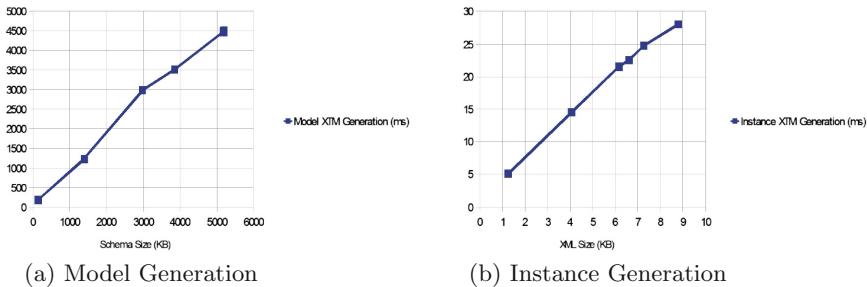
<sup>1</sup> Available at <http://www.w3.org/XML/Binary/2005/03/test-data/UBL-1.0/>.

**Table 2.** Performance Measures for UBL Dataset

Dataset	Schema Size (KB)	Model XTM Generation (ms)	XML Size (KB)	Instance XTM Generation (ms)	XTM Size (KB)
DespatchAdvice	3846.74	3509	6.16	21.54	5629.18
Invoice	1393.73	1228	8.8	28.03	2439.86
Order	5182.27	4481	6.61	22.53	8238.43
OrderCancellation	147	183	1.23	5.15	239.46
OrderChange	5186.51	4509	7.25	24.7	8190.23
OrderResponseSimple	147	185	1.26	5.14	240.82
ReceiptAdvice	2974.95	2987	4.07	14.55	4398.74

## 6 Case Study: Freight Billing Process

Let us now describe a case study of a simple UBL process for illustration purposes, - Freight Billing process given in Fig. 4 which is bereft of any gateways. In this process diagram, while Send Freight Invoice and Receive Freight Invoice are Activities, Freight Invoice is a document interchanged between actors, Accounting Supplier and Accounting Customer.

**Fig. 3.** Graph of performance measures from UBL Dataset

For this Topic Map model; Event, Activity, Sequence and Gateways are modeled as follows. There is a main topic Freight Billing Process which is connected to topics; Actor, Event, Artifact and Activity through ‘contains’ association. Consider an example, where activity ‘Send Freight Invoice’ is performed by ‘Accounting Supplier’. In the Topic Map we represent it by drawing an arrow from activity ‘Send Freight Invoice’ to the actor ‘Accounting Supplier’ through an association ‘assigned-To’. Next we introduce association class named ‘doc-Transfer’ with ‘producingActivity’, ‘consumingActivity’ and ‘doc’ as association roles. In this process, activity ‘Send Freight Invoice’ plays the role of ‘producingActivity’, ‘Receive Freight Invoice’ the role of ‘consumingActivity’ and ‘Freight

Invoice’ the role of ‘doc’ in ‘docTransfer’ association. Moreover, the topic ‘Freight Invoice’ becomes the root element of the schema ontology through *contains* association. The resulting Topic Map is shown in Fig. 5.

For querying Topic Map, a query language called Tolog is used which is similar to Datalog (Horn-clause fragment of Logic Programs) and SQL. For example we pose the following queries on the Topic Map for Freight Billing Process.

- Q1.** List all activities in the process: instance-of(\$TOPIC, activity)?
- Q2.** Is activity Receive Freight Invoice followed by Send Freight Invoice: doc-Transfer(Receive Freight Invoice:producing Activity, Send Freight Invoice: consuming Activity)?

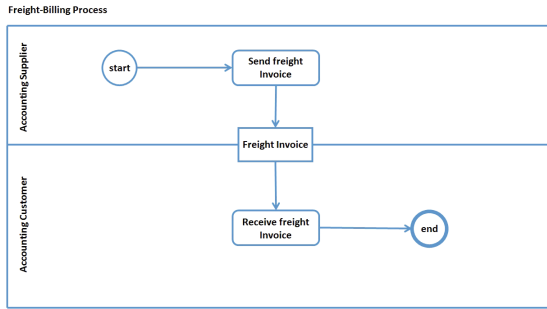


Fig. 4. Freight Billing Process

For each process in UBL we can extract an ontology out of it and create a repository of ontology for all of them by using a root topic element *UBLProcess* which would connect all processes. For generating transitive relationships, inheritance relationship and the like, among processes we can design some inference rules using Tolog, a sample of rules is presented below.

- progeny-Of(\$E11, \$E12) :- subclass-Of(\$E11,\$E12).
- progeny-Of(\$E11, \$E12) :- subclass-Of(\$E11,\$E13),progeny-Of(\$E13, \$E12).

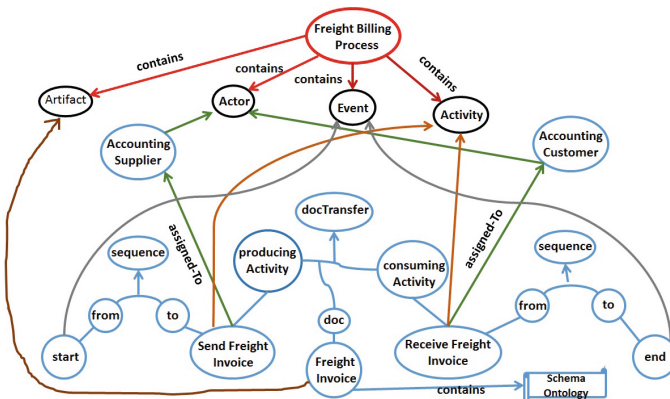


Fig. 5. Topic Map of Freight Billing Process

## 7 Conclusion

Our work can be seen as model-based ontology development [2] of SOE which will of be interest to KBE community as it is trying to embrace a methodological approach on the lines of software engineering with knowledge as its main focus. This approach advocates methods and techniques for knowledge acquisition, modeling and representation, and proposes method of extracting ontology automatically from models commonly used in software engineering such as business processes. Such kind of ontology creation will be useful for development and maintenance of complex KBE-based applications of SOE.

## References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* (2004)
2. Gasevic, D., Djuric, D., Devedzic, V.: MDA-based automatic OWL ontology development. *STTT* **9**(2), 103–117 (2007)
3. Gong, R., Li, Q., Ning, K., Chen, Y., O’Sullivan, D.: Business process collaboration using semantic interoperability: review and framework. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) *ASWC 2006*. LNCS, vol. 4185, pp. 191–204. Springer, Heidelberg (2006)
4. Groener, G., Staab, S.: Modeling and query patterns for process retrieval in OWL. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) *ISWC 2009*. LNCS, vol. 5823, pp. 243–259. Springer, Heidelberg (2009)
5. Hauser, R., Friess, M., Küster, J.M., Vanhatalo, J.: Combining analysis of unstructured workflows with transformation to structured workflows. In: *10th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006)* (2006)
6. Heravi, B.R., Bell, D., Lycett, M., Green, S.D.: Towards an ontology for automating collaborative business processes. In: *Proceedings of the 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops, EDOCW 2010*, pp. 311–319. IEEE Computer Society (2010)
7. ISO/IEC 13250. Topic Maps (2002). [http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322\\_files/iso13250-2nd-ed-v2.pdf](http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf)
8. Karjekar, F., Roy, S., Padmanabhuni, S.: Intelligent business knowledge management using topic maps. In: *Proceedings of the 2nd Bangalore Annual Compute Conference*. ACM (2009)
9. Lacoste, D., Sawant, K.P., Roy, S.: An efficient XML to OWL converter. In: *Proceedings of the 4th Indian Software Engineering Conference (ISEC)*, pp. 145–154. ACM, New York (2011)
10. Legner, C., Wende, K.: The challenges of inter-organizational business process design - a research agenda. In: *ECIS*, pp. 106–118 (2007)
11. Nitzsche, J., Wutke, D., Lessen, T.V.: An ontology for executable business processes. In: *Proceedings of Workshop on Semantic Business Process and Product Lifecycle Management (SBPM 2007)*, vol. 251. CEUR Workshop (2007)
12. Rath, H.H.: *Topic Maps Handbook*. Empolis, Gütersloh (2003)
13. *The Ontology Forum: UBL Ontology* (2003)

14. Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow patterns. *Distrib. Parallel Databases* **14**(1), 5–51 (2003)
15. W3C Recommendation: Universal Business Language v2.0 (2006). <http://docs.oasis-open.org/ubl/cs-UBL-2.0/UBL-2.0.html>
16. Wu, B., Li, L., Yang, Y.: Ontological approach towards e-business process automation. In: *Proceedings of the IEEE International Conference on e-Business Engineering, ICEBE 2006*, pp. 154–161. IEEE Computer Society (2006)
17. Yarimagan, Y., Dogac, A.: A semantic-based solution for UBL schema interoperability. *IEEE Internet Comput.* **13**(3), 64–71 (2009)