Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2017 Proceedings

Mediterranean Conference on Information Systems (MCIS)

9-2017

Transformation From Business Process Models To Process Ontology: A Case Study

Ahmet Coskuncay
Bilgi Grubu Ltd, ahmet.coskuncay@gmail.com

Ozge Gurbuz
Middle East Technical University, gurbuzozge@gmail.com

Onur Demirors
Izmir Institute of Technology, demirorso@gmail.com

TRANSFORMATION FROM BUSINESS PROCESS MODELS TO PROCESS ONTOLOGY: A CASE STUDY

Research full-length paper

Track N°8

Coskuncay, Ahmet, Bilgi Grubu Ltd, Ankara, Turkey, ahmet.coskuncay@gmail.com

Gurbuz, Ozge, Middle East Technical University, Ankara, Turkey, gurbuzozge@gmail.com

Demirors, Onur, Izmir Institute of Technology, Izmir, Turkey, demirorso@gmail.com

Abstract

Business process modeling is utilized by organizations for defining and reengineering their business processes. On the other hand, ontologies are developed to strengthen shared understanding between people, organizations and software systems and ease reuse. From knowledge management point of view, both are efficient tools for creating knowledge. A tool supported transformation from process models to ontology could enhance the benefits gained from both and increase development efficiency and consistency. This study aims to demonstrate such an automated transformation on a real case. Within the study, a case study is performed to enable this transformation manually from business process models defined with eEPC language to a process ontology and an algorithm is designed and implemented for automated transformation.

Keywords: Business process modeling, eEPC, ontology languages, process modeling notations, process ontology.

1 Introduction

Business process modeling is used for a variety of purposes, including, but not limited to, establishing an execution consistency, optimization, automation, measurement and certification. Business process modeling seeks standardization in business process management, where the related business processes might include several applications, data repositories, corporate departments or even companies (Minoli, 2008).

Ontologies, on the other hand, enable people and software reach a shared understanding of the structure of information, reuse of domain knowledge, making domain assumptions explicit, differentiating domain and operational knowledge, and analysing domain knowledge (Gruber, 1993). Ontology development is an essential part of Knowledge Management domain in terms of creating domain knowledge. Similarly, business process models can be regarded as important for organizations to create formal knowledge (Kalpic and Bernus, 2006) from knowledge management perspective. Thus, both activities are utilized as part of knowledge creation.

Domain ontologies can be adapted as process ontologies, which are specifications of the classes and their relations contained in processes and a formal representation of the domain of processes. Process ontologies can incorporate an extensive knowledge about an organization’s processes, activities, roles, application systems, process interfaces, inputs and outputs, which are usually necessary information for creating domain ontologies.

There are studies, which have focused on integrating ontologies with process modeling in order to improve data representations and querying options in semantics level. They define framework for semantic business process management and automate process ontology population from process descriptions and annotations in theoretical level. Furthermore, Mendling et al. (2014) states the importance of process models bounded with the semantics and defines the 25 challenges, which identify the gaps in this research area. Yet, challenge 24 in Mendling et al. (2014) points out that there are fewer studies for discovering ontologies from process models.

Additionally, our experiences has showed that organizations performing both process modeling and ontology building activities, allocate duplicated efforts for each activity since they are using same or similar knowledge resources. Moreover, neither activity benefits from the knowledge created in the other, thus the resulting products are inevitably inconsistent with each other in both creation and maintenance. Therefore, for ontology population using business process models, we propose an alternative approach. We believe that creating process ontologies as a bridge between business process models and domain ontologies can improve efficiency and consistency in development and completeness of work products. Thus, domain ontologies developed by reusing information in business process models would be consistent with processes. Besides, domain analysis, which is a part of ontology building, would benefit from the analysis of processes.

In this study, process ontologies are created by utilizing business process models at first place by performing a case study on a real scenario. According to the case study results and evaluation of the process ontology by the process owners, the manual transformation is validated. Afterwards the transformation tool is implemented. The tool uses business process models that are modeled with a well-established modeling language, namely extended Event-driven Process Chain (eEPC) and produces process ontology defined with RDF. The populated ontology is then published in the web with a graphical representation for enabling the information to be accessible and to be queried visually by the organization. The rest of the paper is organized as follows: the second part gives information about the related work, the third section describes the case study conduct, and the last section concludes with the results, discussion, limitations and future work.

2 Related Work

Research on the relations between process models and ontologies has increased in recent years. Some of the studies (Santos Jr et al., 2010; Davis et al., 2004) investigate process modelling languages based on foundational ontologies and there are some other studies (Höfferer, 2007; Haller et al., 2008; Sönmez et al., 2010) that highlight the importance and uses of process ontologies in practice, though none of these focus on a transformation approach in order to identify how to create ontologies by using process models.

Hepp and Roman (2007) uses semantics in business process management for machine reasoning and intelligent querying. The aim is to represent business process models semantically and therefore they define an outline for representational requirements, ontologies and scope of these ontologies by using competency questions. They define a framework for semantic business process management but they do not provide a transformation. De Cesare, Juric and Lycett's work (2014) also relates process ontologies with process modeling. In their study, a transformation from process models to process ontology is provided, but their study differs from ours in terms of the source for transformation. They populate the process ontology by using the textual descriptions of business process models whereas we populate the ontology from the process nodes and elements. Leopold et al. (2015), on the other hand, is another example for relating ontologies with processes. They automatically generate an annotation taxonomy for process models. Their intent, however, is to match the similarity between process activity with a concept and process model with taxonomy category. In the automation phase for matching annotation to process models, they use Markov Logic formalization. Belhajjame and Brambilla (2009) present ontology usage for business process discovery, querying and modeling. They populate business process ontology concepts using semantic annotations of business process sets and activities. Gurbuz and Demirors (2017a), on the other hand, propose a method for developing process ontologies from scratch using organizational guidelines and establishing process models.

Ontologies for process models are also provided in (Belecheanu et al., 2007; Cimpian et al., 2008) based on languages like EPC, BPMN and BPEL and process model to ontology transformation are described for Petri Net (Koschmider and Oberweis, 2005), BPMN (Di Francescomarino et al., 2009; Francescomarino, 2011; Eisenbarth, 2013) and lean EPC (Thomas and Fellmann, 2009; Eisenbarth, 2013) languages. Our study has similarities with these in terms of transformation approach but differentiates itself in terms of modeling language choice, coverage of several interrelated process models and model elements in ontology and research method. The transformations by Thomas and Fellmann (2009) and Eisenbarth (2013) are from lean EPC (with only workflow elements) to an ontology including general use ontology concepts such as node (where functions and events are mapped) and graph (where process models are mapped). Our study, on the other hand, aims to reach a wider model element set in extended EPC language, which provides the audience more information about the process as compared to lean EPC.

Moreover, we first perform a case study in order to explore how to transform process models to process ontologies on a real case. We, then compare the resulted process ontologies with the process models with domain experts and process owners. Making sure at first place that this method populates process ontologies, which are consistent with the models, we automate the transformation. In addition, we differentiate our work by addressing following issues in mapping process models to ontology:

- Usually, there are multiple instances of the same model element objects scattered across different processes. Such objects need to be represented as a single resource in process models and ontology.
- Processes have sub-processes and interfaces to other processes represented in business process models. These relations are also needed to be represented in process ontology.
- Process modeling languages such as EPC can be extended with model element types. Changing set of model element types used within modeling languages is needed to be easily integrated with the process ontology.

3 Case Study

3.1 Case Study Design and Planning

This case study have two goals. The first goal is to practice and learn from developing an ontology by using the business process models. The second goal is to use the results of the case study in the transformation tool development. The aim is to demonstrate the need for a transformation tool and prototyping the possible inputs and outputs of it. The following research question was characterized: How can the information in business process models be transferred to an ontology?

Following activities were planned to be performed in the case study. First three bullet points refer to case study execution whereas latter three are case study analysis stages of performing a case study (Jedlitschka and Pfahl, 2005).

- Process definition: Business process modeling will be performed for the selected case with chosen modeling language. The resulting models will incorporate process related information such as processes, activities, roles, inputs and outputs.
- Ontology definition: A process ontology, which is compatible with the business process models, will be built. Processes and their elements, both as classes and their instances, are to be represented.
- Process model versus process ontology comparison: Business process models and process ontology would be compared for identifying the representation capability gap. This step is also used for validation purposes.
- Automation potential identification: Automation potential will be evaluated based on the resulting business process models and process ontology.
- Algorithm to demonstrate automation potential: By using the outputs of the case study, automation potential will be demonstrated via a transformation algorithm.
- Transformation tool implementation: Based on transformation algorithm, tool will be implemented.

Three case selection criteria were determined. The first criterion is that the selected case should be ecologically valid. Thus, case study results would be validated by resolving a real-world problem in a real setting. The second criterion is about whether the process owners’ commitment is established for contributing as information providers and validators. Such involvement often necessitates some motivational factors like willingness of business people and continuing management support. The last criterion is that the selected case should incorporate use of a wide range of process element types and perspectives for improving the validity of the study.

Our case study, as selected with respect to the case selection criteria, is part of a project whose scope is bounded by the service of planning and monitoring investments of public agencies. This service, which is performed mainly by Ministry of Development, includes the following processes:

- Determine proposal ceilings for public organizations,
- Give visa to allocations,
- Publish investment program,
- Gather investment project details,
- Revise investment program,
- Monitor and evaluate investment program,
- Finalize investment projects selected.

3.2 Case Study Execution and Analysis

In any organization, setting the scope for ontology building is critical since, related engineering activities can easily deviate from their purposes. Organizations might incorporate various interrelated domains, functional divisions and services, which might expand the intended scope with unintended additions. This issue can be resolved by managing the scope of the study from start to the end. A possible solution as offered by the METHONTOLOGY (Fernández-López et al., 1997) is setting goals that govern scope of the study.

Specifying competency questions upfront is also useful for defining the scope. Competency questions are the questions that resulting ontology is expected to provide answers to (Uschold and Grüninger, 1996). Querying the ontologies with respect to competency questions is used for verification and validation of ontologies in many studies. Using competency questions is common in some well-established methodologies such as METHONTOLOGY (Fernández-López et al., 1997) and NeOn (Suárez-Figueroa, 2010). Ontology development goal and related competency questions are provided in Table 1.

Goal Definition: Making knowledge about investment program preparation processes explicit.
Goal Description: As per the law concerning the establishment of it, Ministry of Development is responsible from establishing public investment politics, performing analysis and research about investments, supporting development of public investment project ideas, analyzing projects, preparing, monitoring and evaluating investment program. However, the processes for preparing and managing them are mostly implicit. Preparation processes are intended to be made explicit to be queried by using multiple parameters and integrated and annotated with domain knowledge later on.
Competency Questions: Which roles are responsible for publishing investment program? Which activities are in responsibility of each role? By which role and in which processes can investment program be revised? Which activities are automated by which application systems? Which work products that are output of a process are input to another process? Which end states exist in each process? Which external processes exist in preparing investment program? In which process and by whom a certain work product is modified? …

Table 1. Definition of ontology development goal.

Modeling "publish investment program" process is determined to be within the scope of the case study. Although, all seven processes above were planned to be modelled (and were modeled as planned indeed), only "publish investment program" process was selected to be utilized in populating the ontology with individuals. "Publish investment program" process was selected since it corporates several roles, input and outputs and applications, which serves our purpose to demonstrate process ontology individuals of various classes.

Extended Event-driven Process Chain (eEPC) provides strong analysis capabilities of business domain, behavioral, information and organizational perspectives with the set of model elements included in it. It has a rather flexible notation (except for lean EPC model elements such as function, event and logical connectors, which are obligatory for work flow perspective) and thus could be further extended or narrowed in terms of model elements to be used. This is why selecting the modeling language should also cover selecting the model elements and modeling rules. Following model elements were selected to serve the modeling purposes agreed with process owners: function, event, logical connectors (AND, OR, XOR), information carriers, application, organizational elements, process interface, improvement offer, and technical term.

Knowledge about processes can exist in various representations and formats, even if it is not defined in process models. It might be specified in regulatory documents, automated within information systems, or owned by the business people as tacit knowledge. Usually process modelers elicit process data using these sources while modeling the processes. The set of processes within the scope of planning and monitoring investments were modeled with collaboration of the process analysts (modeling specialists that are external to the organization) and process owners (domain experts within the organization). More explicitly, the process analysts, at first step, prepared draft process models by using regulatory documents to ensure that documented knowledge about the processes is represented in the process models. Then in several iterations, draft models were walked through with the process owners in workshops and revised by the process analysts. Business process model for "publish investment program" process is shown in Fig.1, as validated via walkthroughs and reviews by the process owners.

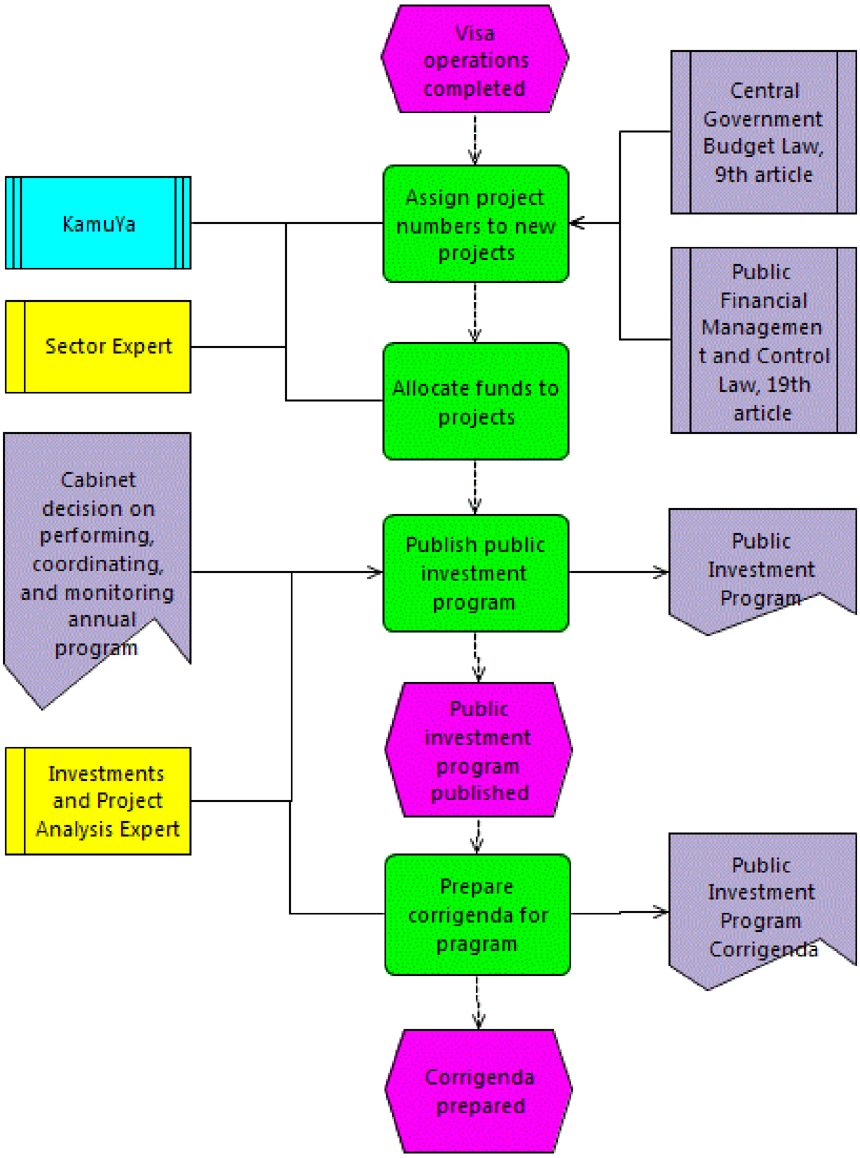

Figure 1. Business process model for publish investment program process.

As a next step for defining process ontology, a concept diagram for business process model conforming to eEPC modeling language was designed. Basically the concept diagram is a representation of metadata model of the selected modeling language. It includes classes for model elements and the process itself. Information flow, work flow, process responsibility and generalization relations were also defined between classes based on the relations in eEPC. We omitted some model elements (i.e. logical connectors (AND, OR, and XOR)) and relations (i.e. function-to-event relation) in eEPC language as their presence in ontology do not make much sense from the user point of view as evaluated in workshops with the customer. The concept diagram is provided in Fig.2. A process ontology, which defines process-related classes and relations, was implemented by using the concept diagram, whose sample is provided in Table 3. The instances for model elements were taken as individual candidates and related to the process ontology classes. On ontology editor, individuals were created for ontology classes and related object properties and data properties were defined respectively before the ontology was peer reviewed for verification. A sample of the output is given in Table 4.

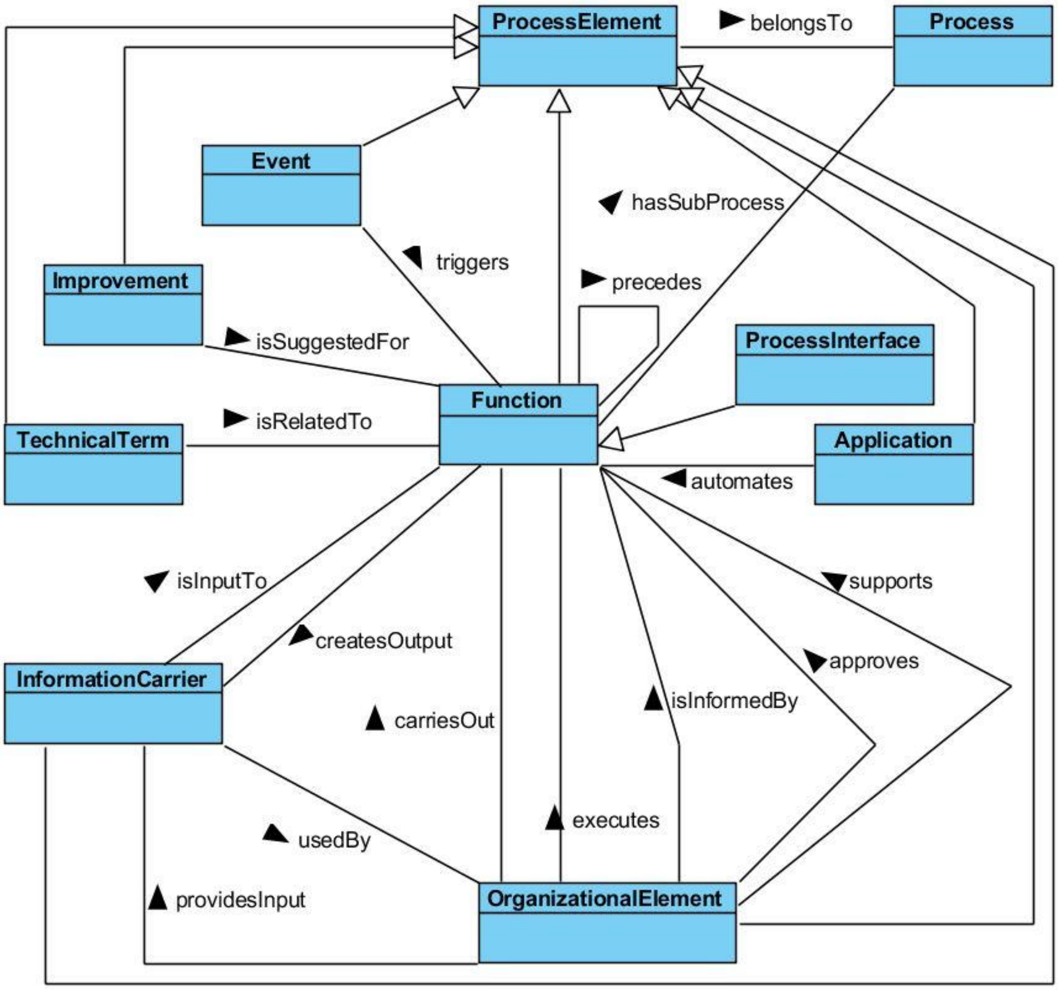

Figure 2. Concept diagram based on eEPC modeling language.

3.3 Case Study Results

As we compared business process models and process ontology, it was clear that all the information contained in business process models were included by the process ontology. On the other hand, process ontology brings formalism and semantic verification opportunities to business processes when compared to the business process models. Axioms that constrain types of process elements and relations between process elements enable these opportunities; as such semantic properties are usually expected to be provided by modeling tools.

Resulting process ontology included fourteen classes, twenty-two object properties and sixteen individuals. A graphical representation of it was formed and walked through with process owners.

As process ontology and business process models were consistent and compatible in terms of process elements and their instances, potential for automated transformation was conceived to be high. This potential was also evaluated by mapping XMI codes (business process model information) to RDF codes (process ontology) for explanatory purposes. We achieved a complete mapping that validates the automation potential.

3.4 Automation of Ontology Population

As result of the case study, it is seen that populating ontology from process elements and nodes creates consistent process ontology, which fully covers the business processes. After being sure of this way of populating ontology is a valid method, we defined the algorithm for transforming business process models, which were defined by the process modelers, to ontology. This transformation algorithm is designed in order to validate the transformation potential. More explicitly, its aim is to show the possibility for utilizing XMI files representing business process models and producing RDF files of process ontology.

The transformation process is composed of three parts. The first part is reading the XMI documents and creating individuals of each element. The second part is reading the high level process ontology model and creating the new ontology model in which the individuals will be added. The last part is merging the process element individuals to ontology classes in the new model. To further describe transformation from business process models to process ontology, samples of input and output of the transformation are provided in Table 3 and 4 respectively. Due to illustrative purposes, only two model elements and relations between these two are sampled from Fig.2 to be used in this section.

The tool used for modeling our processes (i.e. UPROM) stores model information as XMI specifications. Each element in the XMI has an ID, name and type. On the other hand, each connection in the XMI has an ID, type, source and target elements' IDs. In order to illustrate the flow in the diagram, our first step is formalizing the elements with connections into a source/target form. We also, however, define the individuals as source/target form in this step. Pseudo code below shows the method:

```
for each connection in connectionList
        for each element in elementList
                if(connection.source==element)
                        sourceTarget.source=element;
                else if(connection.target==element)
                        sourceTarget.target=element;
                if(sourceTarget.notEmpty())
                        sourceTargetList.push(sourceTarget);
                        indvList.source = newModel.createIndividual(sourceTarget.source);
                        indvList.target = newModel.createIndividual(sourceTarget.target);
```

This pseudo code creates the following sample outputs:

source: Visa operations completed – target: Assign project numbers to new projects

source: KamuYa – target: Assign project numbers to new projects

The second step is to create a new model by reading meta-model of process ontology. This high level ontology is externally managed to handle changes in modeling language. Our transformation algorithm uses this high level process ontology in order to create a new model. We read the model using Apache Jena libraries. First, we read the statements, get their subjects and get their properties. Afterwards we convert it to more useful form: subject-predicate-object as shown in the pseudo code below:

```
for each statement in loadedModel.listStatements()
        subject= statement.getSubject()
        for each property in subject.listProperties()
                list.subject= property.getSubject();
                list.predicate= property.getPredicate();
                list.object=property.getObject();
                subjectList.push(list);
for each list in subjectList
        for each list2 in subjectList
                if(list.subject == list2.subject&&list.predicate == domain && list2.predicate
                == range)
                                node.subject=list.object;
                                node.predicate=list.subject;
                                node.object=list2.object;
                                nodeList.push(node);
```

Afterwards this loaded model is used for creating the new model. The pseudo code below shows the creation of classes and properties by using Apache Jena libraries.

```
newModel= ModelFactory.createOntologyModel();
for each node in nodeList
                newModel.createClass(node.subject);
                newModel.createClass(node.object);
                newModel.createObjectProperty(node.predicate);
```

In order to illustrate this algorithm we give the sample high-level process ontology input file defined with Protégé tool in RDF language in Table 3. In this sample file, ontology classes are defined for “Function”, “Event”, and “Process” and object properties are defined for “triggers” and “belongsTo” relationships. They simply indicate the following triples with domains and ranges:

Event – triggers – Function

ProcessElement – belongsTo- Process

Event – isSubclassOf – ProcessElement

Function – isSubclassOf – ProcessElement

The algorithm we defined reads the properties, their domain and range classes and creates the node list according to the property’s domain and range. Then this data is used for creating the classes and object properties of the new model.

The last step in the transformation algorithm is to merge the processes with the defined new ontology model. In this step, we use the node list that is generated from the high-level process ontology and individuals list which is generated from the process elements. The algorithm in this step matches the individuals under their relevant classes with related properties. The pseudo code for this merging is defined as below:

```
for each individual in indvList
        for each node in nodeList
```

if(individual.source.getOntClass()==node.subject&&individual.target.getOntClass() == node.object

Property property= newModel.createObjectProperty(node.predicate);

Individual.source.addProperty(property, individual.target);

The sample from the resulted output is given in Table 4, which is an interpretation of the process ontology input merged with the business process individuals. The simpler way to illustrate the merge result is:

Visa operations –is a- Event

Visa operations – triggers – Assign project numbers to new project

Visa operations – belongsTo – Publish Investment Program.epc

Assign project numbers to new project – is a – Function

Assign project numbers to new project – belongsTo – Publish Investment Program.epc

Publish Investment Program.epc – is a – Process

Final step was to implement the transformation algorithm. Transformation tool was implemented by using Apache Jena RDF API. It takes XML files including business process model information and RDF file including process ontology that defines classes and properties (i.e. TBox) as input. Resulting RDF triples, which include individuals, classes and properties, are stored in Virtuoso.

In our case study, we used all seven business process models and their sub-processes (i.e. there were two sub-processes) in populating the process ontology and integrating resulting triples with the ontology infrastructure of the organization. Process models to ontology transformation was performed and results were evaluated by reviews. The number of total triples and process element individuals generated by the transformation tool using seven business process models for planning and monitoring public investments as input is presented in Table 2.

Business processes	Total # of triples	Function	Event	Improvement	Organizational element	Technical term	Information carrier	Application
Determine proposal ceilings for public organizations	198	25	9	4	11	5	34	3
Give visa to allocations	85	12	7	1	10	3	8	2
Publish investment program	35	4	3	3	2	0	5	1
Gather investment project details	33	5	3	0	3	3	3	1
Revise investment program	44	9	4	0	5	0	4	1
Monitor and evaluate investment program	92	4	3	8	5	0	28	1
Finalize investment projects	20	2	2	3	2	0	2	0
TOTAL	507	61	31	19	38	11	84	9

Table 2. Summary of business process models to ontology transformation in case study.

These triples are an integrated part of a larger ontology in Ministry of Development. This developed ontology is published on the web and are used for answering questions such as; who is responsible for a certain activity, or who should produce a certain artefact. The resulted process ontology is currently being used by the Ministry of Development. The mock-up (the real screenshot is not in English) of the querying sample results of the organizational element "Sector Expert" and the application system "KamuYa" is given in Fig.3. Integrating process knowledge to organizational knowledge base via merging process ontology to domain ontologies in managing public investments domain enabled an enhanced knowledge base for the organization to query information.

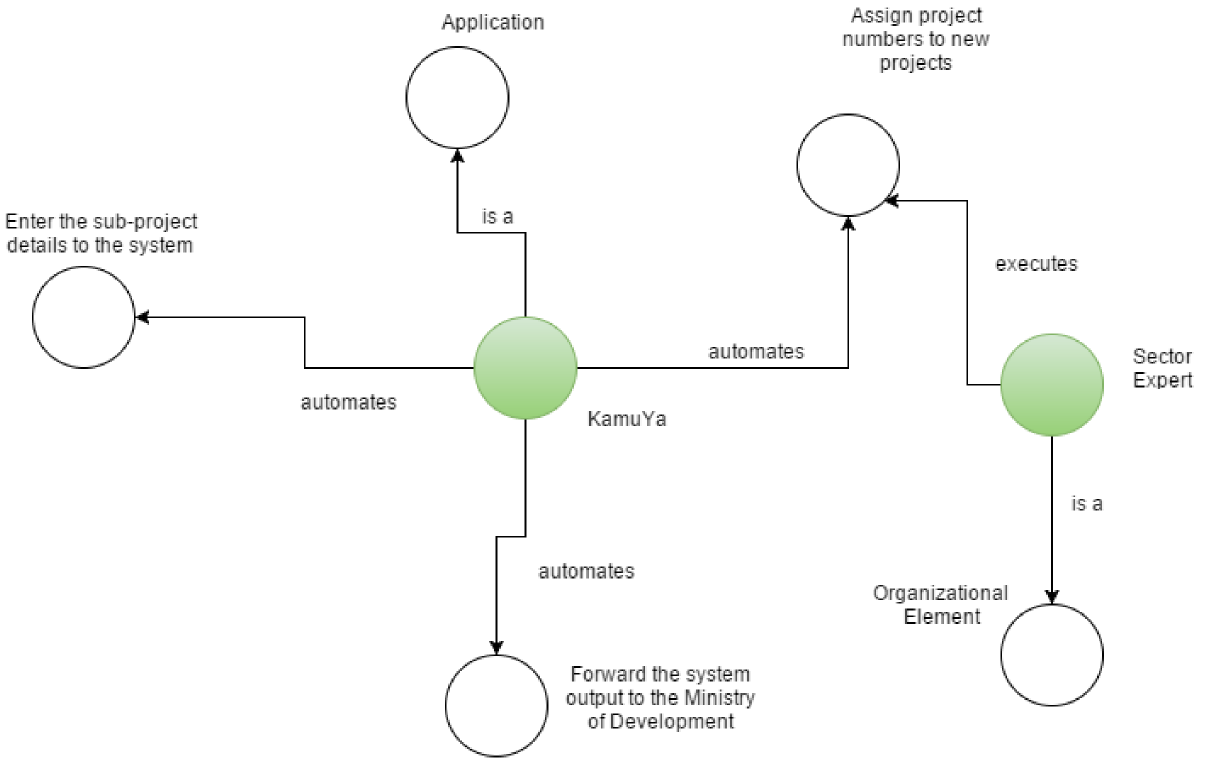

Figure 3. A sample of the visual representation of the resulted ontology.

4 Conclusions, Limitations and Outlook

Organizations usually perform business process modeling and ontology development as separate activities, which share some similar analysis tasks and business knowledge. In this study, we present a transformation approach that benefit these shared tasks and knowledge and increase the efficiency in development. Our case study utilized business process models in transformation to process ontology. We present a process ontology based on eEPC language and tool to enable such a transformation. We envision an increase in consistency with business processes and decrease in analysis effort as the resulting ontology elements will be used in developing domain ontologies. In this study, the potential is demonstrated for transformation from the process models modeled with eEPC to the process ontology that would be used in building domain ontologies. In addition, the process ontology designed in our study enables such transformation for sets of business processes that includes several processes interacting with each other as we see them in most ecologically valid cases. Moreover, as eEPC is needed to be further extended (e.g. by adding or deleting a model element type), modifying process ontology classes simply keeps the transformation work as expected.

The outputs of the performed case study include the business process models for planning and monitoring public investments, an ontology that includes fourteen classes and twenty-two object properties between classes that represent the process, process elements and relationships between them and lastly a transformation tool. The resulting process ontology is an integrated part of a larger domain ontology in the focus organization. The transformation tool is capable of producing ontology constructs that would be used in developing related domain ontologies.

One of the limitations of the study is that the transformation presented is process modeling language (i.e. eEPC) dependent. If another modeling language or different model elements are selected by practitioners, our high level process ontology needs to be modified or redesigned with an ontology editor in order to make the transformation algorithm work. Other tools to be developed based on our algorithm would also be dependent on the language used in the process model input, which affects process modeling tool selection.

Another limitation is that our work aimed for one-way transformation from process models to ontology. Thus, currently changes in ontology cannot be reflected back to process models. This might prevent practitioners to modify the ontology or annotate it with domain concepts, since as process models change; our algorithm overwrites the process ontology without keeping modifications on it. Seeking potentials for maintaining process models and ontology without sacrificing consistency and enabling two-way transformation are in our short-term future research agenda.

Outputs of this study support ontology building by using business process models. Our future plan is to integrate this work with the method and tool support for integrated development of business process models and domain ontologies that are compatible in creation and maintenance (Coskuncay et al, 2017), thus bringing benefits including improvement in interoperability of organizations. Additionally, to extend the output of this study, integrating the established ontology with the process ontologies developed from the organizational guidelines (Gurbuz and Demirors, 2017b) is in our agenda.

<ObjectProperty rdf:about="http://www.semanticweb.org/ ontologies/2014/11/processOntology#triggers"> <rdfs:domain rdf:resource="http://www.semanticweb.org/ ontologies/2014/11/processOntology#Event"/> <rdfs:range rdf:resource="http://www.semanticweb.org/ ontologies/2014/11/processOntology#Function"/> </ObjectProperty>
<ObjectProperty rdf:about="http://www.semanticweb.org/ontologies/2014/11/processOntology#belongsTo"> <rdfs:range rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#Process"/> <rdfs:domain rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#ProcessElement"/> </ObjectProperty>
<Class rdf:about="http://www.semanticweb.org/ontologies/2014/11/processOntology#Event"> <rdfs:subClassOf rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#ProcessElement"/> <rdfs:subClassOf> <Restriction> <onProperty rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#triggers"/> <onClass rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#Function"/> </Restriction> </rdfs:subClassOf> </Class>
<Class rdf:about="http://www.semanticweb.org/ontologies/2014/11/processOntology#Function">

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#ProcessElement"/>
</Class>
```

```
<Class rdf:about="http://www.semanticweb.org/ontologies/2014/11/processOntology#ProcessElement">
<rdfs:subClassOf>
<Restriction>
<onProperty rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#belongsTo"/>
<someValuesFrom
rdf:resource="http://www.semanticweb.org/ontologies/2014/11/processOntology#Process"/>
</Restriction>
</rdfs:subClassOf>
</Class>
```

```
<Class rdf:about="http://www.semanticweb.org/ ontologies/2014/11/processOntology#Process"/>
```

Table 3. Sample RDF declarations for three classes and two object properties.

```
<http://www.semanticweb.org/ontologies/2014/11/processOntology#VisaOperationsCompleted>
a       <http://www.semanticweb.org/ontologies/2014/11/processOntology#Event> ;
rdfs:label  "Visa operations completed" ;
<http://www.semanticweb.org/ontologies/2014/11/processOntology#belongsTo>
<http://www.semanticweb.org/ontologies/2014/11/processOntology#PublishInvestmentProgram.epc> ;
<http://www.semanticweb.org/ontologies/2014/11/processOntology#triggers>
<http://www.semanticweb.org/ontologies/2014/11/processOntology#AssignProjectNumberstoNewProject > .
```

```
<http://www.semanticweb.org/ontologies/2014/11/processOntology#AssignProjectNumberstoNewProject >
a       <http://www.semanticweb.org/ontologies/2014/11/processOntology#Function> ;
rdfs:label  "Assign project numbers to new project" ;
<http://www.semanticweb.org/ontologies/2014/11/processOntology#belongsTo>
<http://www.semanticweb.org/ontologies/2014/11/processOntology#PublishInvestmentProgram.epc > ;
```

```
<http://www.semanticweb.org/ontologies/2014/11/processOntology#PublishInvestmentProgram.epc >
a       <http://www.semanticweb.org/ontologies/2014/11/processOntology#Process> ;
rdfs:label  "Publish Investment Program.epc" .
```

Table 4. Sample output from the transformation tool.

References

Belecheanu, R., L. Cabral, J. Domingue, W. Gaaloul, M. Hepp, A. Filipowska, M. Kaczmarek, T. Kaczmarek, J. Nitzsche, B. Norton, C. Pedrinaci, D. Roman, M. Stollberg and S. Stein (2007). "Semantics Utilized for Process management within and between Enterprises.".

Belhajjame, K. and M. Brambilla (2009). "Ontology-Based Description and Discovery of Business Processes." Lecture Notes in Business Information Processing 29 LNBIP, pp. 85–98.

Cimpian, E., S. Komazec, D. Lintner, C. Blamauer and M. Evenson (2008). "Business Process Modeling Ontology BPMO final version.".

Coskuncay, A., O. Gurbuz, O. Demirors and E.E. Ekinci (2017). "PROMPTUM Toolset: Tool Support for Integrated Ontologies and Process Models." In: Dumas M., Fantinato M. (eds) Business Process Management Workshops. BPM 2016. Lecture Notes in Business Information Processing, vol 281, pp 93-105.

Davis, I.G., M. Rosemann and P.F. Green (2004). "Exploring Proposed Ontological Issues of ARIS with Different Categories of Modellers." in Proceedings of the Australasian Conference on Information Systems.

De Cesare, S., D. Juric and M. Lycett (2014). "Toward The Automation of Business Process Ontology Generation." Proc. - 16th IEEE Conf. Bus. Informatics, CBI 2014. 1, 70–77.

Di Francescomarino, C., C. Ghidini, M. Rospocher, L. Serafini and P. Tonella (2009). "Semantically-Aided Business Process Modeling." in International Semantic Web Conference (ISWC 2009), pp. 114–129.

Di Francescomarino, C. (2011). "Semantic Annotation of Business Process Models." University of Trento.

Eisenbarth, T. (2013). "Semantic Process Models." University of Augsburg, 2013.

Fernández-López, M., A. Gómez-Pérez and N. Juristo (1997). "METHONTOLOGY: From Ontological Art towards Ontological Engineering." in Proceedings of the AAAI97 Spring Symposium, pp. 30–40.

Gruber, T.R. (1993). "A Translation Approach to Portable Ontology Specifications." Knowledge Acquisition 5, 199–220.

Gurbuz, O. and O. Demirors (2017a). "From Organizational Guidelines to Business Process Models: Exploratory Case for an Ontology Based Methodology." Proceedings of the 19th IEEE Conference on Business Informatics, In-Press.

Gurbuz, O. and O. Demirors (2017b). "A Comparison of Process Ontology Discovery from Organizational Guidelines in Two Different Languages." Proceedings of the 43rd Conference on Software Engineering and Advanced Applications (SEAA), In-Press.

Haller, A., M. Marmolowski, E. Oren and W. Gaaloul (2008). "A Process Ontology for Business Intelligence," Business, no. April.

Hepp, M. and D. Roman (2007). "An Ontology Framework for Semantic Business Process Management." Wirtschaftsinformatik 2007, pp. 1–18.

Höfferer, P. (2007). "Achieving Business Process Model Interoperability Using Metamodels and Ontologies." in Proceedings of the 15th European Conference on Information Systems (ECIS 2007), pp. 1620–1631.

Jedlitschka, A. and D. Pfahl (2005). "Reporting Guidelines For Controlled Experiments In Software Engineering." in 2005 International Symposium on Empirical Software Engineering, ISESE 2005, pp. 95–104.

Kalpic, B. and P. Bernus (2006). "Business Process Modeling Through the Knowledge Management Perspective." Journal of Knowledge Management, vol. 10, no. 3. pp. 40–56.

Koschmider, A. and A. Oberweis (2005). "Ontology Based Business Process Description." in Proceedings of the CAiSE 2005, vol. 5, pp. 321–333.

Leopold, H., C. Meilicke, M. Fellmann, F. Pittke, H. Stuckenschmidt and J. Mendling (2015). "Towards the Automated Annotation of Process Models." Advanced Information Systems Engineering: 27th International Conference, CAiSE 2015, pp. 401–416.

Mendling, J., H. Leopold, and F. Pittke (2014). "25 Challenges of Semantic Process Modeling." International Journal of Information Systems and Software Engineering for Big Companies 1, 78–94.
Minoli, D. (2008). "Enterprise Architecture A to Z: Frameworks, Business Process Modeling." SOA, and Infrastructure Technology. Auerbach Publications Boston, MA, USA.
Santos Jr, P.S., J. Almeida and G. Guizzardi (2010). "An Ontology-Based Semantic Foundation for ARIS EPCs." in Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 124–130.
Sönmez, V.N., M. Canlı, S. Gökçe, M. Ünver and A.N. Güçlü (2010). "Ontology Driven Government Services Inventory and Business Process Management." in Enterprise, Business-Process and Information Systems Modeling: 11th International Workshop, BPMDS 2010, pp. 26–38.
Suárez-Figueroa, M. (2010). "NeOn Methodology for Building Ontology Networks: Specification, Scheduling and Reuse." Universidad Politécnica de Madrid.
Thomas, O. and M. Fellmann (2009). "Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes." Business & Information Systems Engineering, vol. 1, no. 6, pp. 438–451.
Uschold, M. and M. Grüninger (1996). "Ontologies: Principles, Methods, and Applications." In: Knowledge Engineering Review 11, 2, pp. 93-155.