An Ontological Meta-model for Business Process Model and Notation (BPMN)

Anna Lisa Guido *annalisa.guido@unisalento.it*
Department of Engineering for Innovation
University of Salento
Via per Monteroni 73100 Lecce

Andrea Pandurino *andrea.pandurino@unisalento.it*
Department of Engineering for Innovation
University of Salento
Via per Monteroni 73100 Lecce

Roberto Paiano *roberto.paiano@unisalento.it*
Department of Engineering for Innovation
University of Salento
Via per Monteroni 73100 Lecce

Abstract

Business processes are key aspect in the modern companies. Several tools to manage business process allow people to design, deploy and monitor business processes. The problem is that business processes are not part of the company knowledge but they are available only in the business process tool (if there are any) and/or in the mind of designer. In order to introduce the business process design in the overall company's knowledge it is important to represent it in a formal way.

We select BPMN as notation to represent business process and we select ontology as a tool to represent in a machine readable format the business process and to make possible its definition in the company knowledge base.

In this paper we analyze the methodological steps that brings to the ontological meta-model of BPMN 2.0. We will present a detail of classes and properties that made up the ontological meta-model using OWL language. The characteristics of the obtained meta-model are completeness and extensibility. The use of ontological meta-model to represent the BPMN allow to add a semantic layer to the business process design (semantic annotation).

Keywords: BPMN, Ontology, Meta-model, Business Process, Knowledge Base.

1. INTRODUCTION

Starting from the importance of Business Process in the modern companies, it is clear that it is very important both to have a machine readable representation of the business process (that allows to make understandable by the machine the design made up by business expert) and to add and to connect to the company's knowledge base the business process.

In this paper we present an ontological approach to the semantic and machine readable definition of a business process. We select the BPMN[1] as notation to represent the business process. In short we define a meta-model of the BPMN where all the notation details are defined.

The concept of meta-model has been established more than 10 years ago when OMG introduced a meta-model architecture useful to define the UML language [2].

According to [3] “A model is an abstract representation of a system and a meta-model is an abstract description of a model”.

Meta-model are largely used in meta-case tool that allow to separate the notation from the specific methodology different from traditional tool that hide the notation in a code and thus making complex any change, also little change, of the language.

Based on our experience, in order to make effective a meta-model, it must have three characteristics:

- Complete: the meta-model must cover all language primitives;
- Extensible: it is important that the meta-model follows the evolution of the represented language. It is important to have the possibility to extend the meta-model without changing it whenever the language will be modified.
- Semantic: the meta-model must express all the semantic of each primitives of the represented language.

Very important for the meta-model definition is the selection of the language useful to represent meta-model. The language must be machine readable and must make possible to make the meta-model complete and extensible in order to provide the right semantic to each language primitive.

Our selection for the meta-model representation is the ontological language OWL[4]. This selection comes from the semantic expressivity useful to represent all primitives of the meta-model.

There are several experience about the OWL’s use to represent meta-models and they are represented in [5]; some of these are related to web application design methodology. Starting from this technique of meta-model representation the importance to have a structured approach also in the business process design it has come to light. In the past we realized an ontological meta-model of the BPMN 1.2.

This paper focus on the definition of an ontological meta-model for the BPMN 2.0. We highlight how the meta-model answer to the three characteristics previous defined and allow to provide a clear and detailed representation of all design primitives.

The most important improvement of the meta-model definition is related to the use of semantic. Nowadays information systems are more oriented to the gathering of knowledge. Very often the link between company knowledge and business process is only in the mind of the designer. Our research work is a first step through the possibility to include in the company’s knowledge also the business process design in order to use it more and more. The business process design will be enriched using semantic links (semantic annotation) to ontologies that represent different part of company’s knowledge base (for example company chart, law and regulation useful for the company and so on).

In this paper, section 2 describes the state of the art about the use of ontologies for business process design and, starting from the section 3, all the methodological steps to obtain the BPMN 2.0 ontological meta-model will be described. In the section 4 we present a case study that highlight the impact and improvements of the use of semantic meta-model. Finally, in the conclusions and future works, we present our considerations and research direction.

2. RELATED WORKS

The use of ontologies in the business process design is not new in the international scientific panorama.

The area that study from many year the use of ontologies in the business process design is the "Semantic Business Process Management" "(sBPM) [6] and show that business users may have a unified view of the business process (including both models and instance of the model) both in notation and in a machine-readable format. In this way it is possible to make query to these processes. Authors allow to merge Semantic Web Service techniques and BPMN in order to make a new idea under the sBPM name.

The main goal of this research is to reduce the gap between "business perspective" and "actual execution" of the process starting form the idea that in order to obtain the "actual execution" from the "business perspective" there is much manual work that ontologies may reduce.

In the sBPM idea, ontologies are useful to semantically describe the business expert requirements, to semantically describe processes, fragment of processes and IT infrastructures and to make reasoning based on machine reasoning.

The goal is, in fact, to be able to apply machine reasoning for the translation between these two spheres, in particular for the discovery of processes and process fragments and for process composition.

Another method to semantically represent business process is to use OWL-S (OWL for Web Services) [7] used to enrich with semantic information the business process representation.

In [8] authors put in relation services expressed in OWL-S in order to enrich the flexibility offered with BPEL standard used in the business process design.

Soon, these techniques of process modeling, typical of the years 2003-2005 have given way to more sophisticated techniques able to enclose in a single format both the process description and the semantics contained in it.

To this transition, an important role is of the BPMN standard [1] that in a few years became a standard de-facto to design business process.

The attention goes to the study of ontological tools useful to represent business process based on the BPMN notation in the specific, as described in this paper, the attention is in the definition of ontological meta-model useful to define BPMN primitives. There are several proposals oriented in this way and this highlight the importance to have an ontological meta-model of the BPMN notation.

The first BPMN ontology was called the sBPMN ontology [9] developed within the SUPER project and based on the BPMN 1.0. The classes correspond to the elements of BPMN and they are classified in categories like Flow Objects, ConnectingObjects, Swimlanes, Artifacts and Processes. The whole ontology consists of 95 classes and about 50 axioms.

In [10] a BPMN ontology is defined with the dual goals: to provide a knowledge base useful during the analysis of the process and useful to helps in the understanding of BPMN notation and to provide a syntax checker to validate the BPMN model.

Natschläger [10] presented an ontological formalization of BPMN 2.0, built according to similar principles to the ones we applied for modelling our ontology.

Another BPMN ontology was from Oracle [11] with the aim to extend the Oracle BPM ontology by establishing relationships between the OWL classes of the BPMN 2.0 ontology and the OWL classes of the SCA runtime ontology. The Oracle BPMN ontology is made up of:

- Classes and properties that define all BPMN 2.0 elements
- Classes and properties useful to create the specific instance of the process and thus to cover all runtime aspects.

Another proposals of ontology formalizing BPMN are mentioned in [12].

Not all the ontologies are available for download when this paper is writing (May 2016). The already existing ontologies are very complex and, often, the main BPMN concepts are not highlighted. This make complex the implementation of the ontology in a tool for business process editing.

For these reason in this paper we formalize a more simple ontology in order to represent the BPMN notation.

3. THE BPMN META-MODEL

The definition of BPMN ontological meta-model follows an iterative process that allow to obtain the meta-model through stepwise refinement and it is based on the Stanford University methodology [13].

The steps of the process are:

- Definition and representation of the main BPMN notation using ontological classes (ex BPMN_ element);
- Definition of more specific concepts of the BPMN 2.0 and representation of the classes as subclasses of the classes defined in the previous step (ex. graphical_ element);
- Definition of concept to support the design (additional elements) and useful to create the ontological class in the meta-model (ex. supporting_element);
- Organization of classes defined in the previous step in a class/subclass hierarchy;
- Definition of the Object Properties: definition of attributes, restrictions and relationship between concepts useful to create semantic relationships between concepts;
- Definition of instance of the concepts;

In each of the previous step we use the reasoner in order to check the model consistency.

Inconsistency problems must be resolved at each step before to go on next step.

The OWL constructs used in all step of the meta-model design was: intersection, union, complement, subClassOf relation and restriction about property.

The meta-model consists of:

- Disjoint classes: (defined as assertion) that allow to make a concept hierarchy of concepts that represent primitives;
- Classes defined from instances: they represent object of a classes (enumeration), used to provide a detail of each primitive (es. task_type that represent several type of task);
- Classes defined as restriction about properties in order to detail each element in a specific type of element (ex. event_end).
- Properties that allows to put in relationship the element of classes.

Classes are both abstract and concrete. Abstract classes are container of other classes and it isn’t possible to create instances of these classes. Concrete classes are those where it is possible to define instances of BPMN design.

3.1 Meta-model Overview – The Set of Classes of The BPMN Ontology

The main class defined in the meta-model is BPMN_element. This class is composed of two disjoined subclasses:

- graphical_element that contains all the main primitives used to model the business process such as flow_object, connecting_object; swimlane, artifact and data. Each primitives is modelled as a single subclass;
- supporting_element that groups all the additional elements useful to model the attributes of the graphical_element;

The figure 1 shows the meta-model classes in the graph structure.

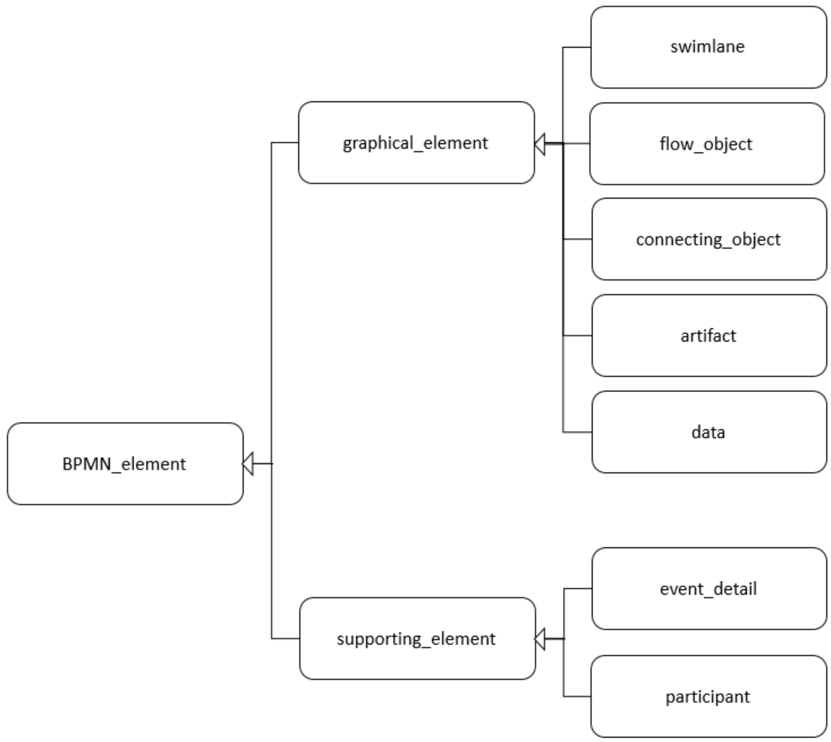

FIGURE 1: Main Subclasses of the BPMN_element.

In the detail, the subclasses swimlane is the generalization of the lane and pool class as designed in figure 2:

Pool is the graphical representation of a subject or organization that takes part in the processes. The pool class acts as a container of the all tasks in which the subjects or organizations are accountable;lane models the tasks inside the pool that a specific actor can do. It allows to organize and to divide the tasks.

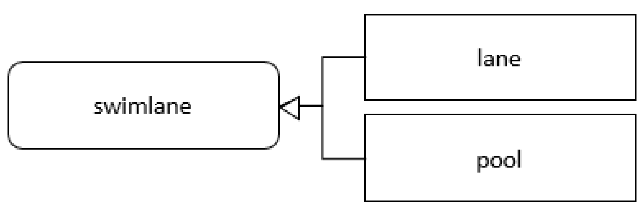

FIGURE 2: Swimlanes Components.

The subclass flow_object models the BPMN primitives that describes the behavior of the business process. The flow object is specialized in the following subclasses:

- activity that has the aspect of a rectangle with smooth angles. It describes a generic activity that could be single or articulated. Its types are task and sub-process.
- Gateway that allows to control the points of union of several flows or the point in which a single flow is spited in several flows.
- event that models what happens during the progress of the process. It describes the flow of the process and usually has a trigger and a result. The events are specialized on the basis of their role in process flow: start, intermediate and end;

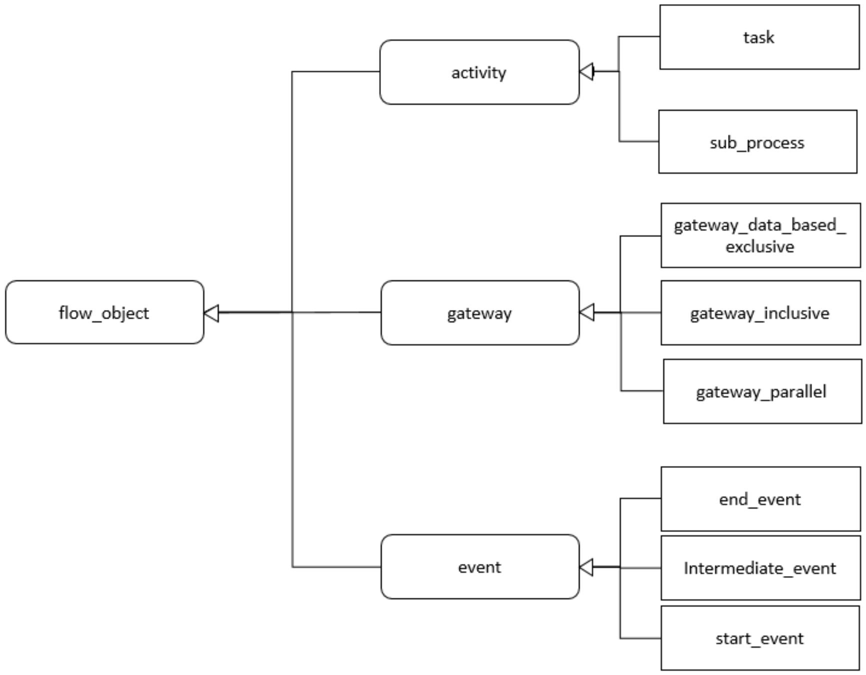

FIGURE 3: flow_object class.

The subclass connecting_object describes all the elements that model the connection flow, the message flow and the association flow, Thus, the class is specialized in sequence_flow, message_flow and association (Figure 4).

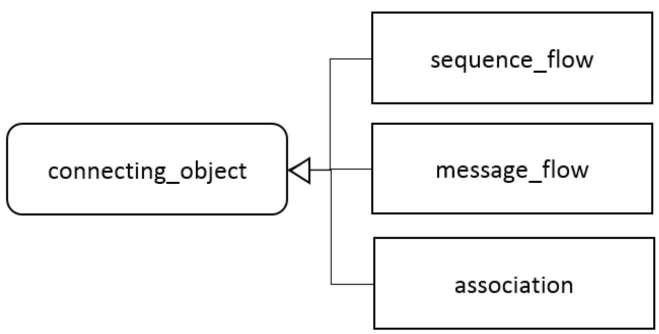

FIGURE 4: connecting_object class.

The artifact class contains the annotations (that allow process designer to add notes) and group (that is a logical artifact that allows to arrange all the activities of same category). Annotation and group do not affect the process flow (fig. 5).

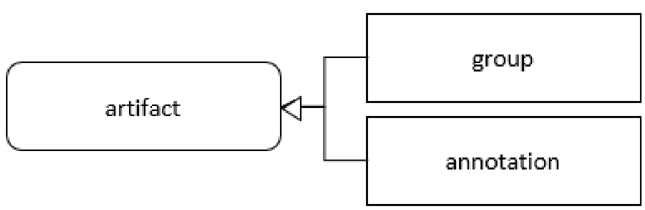

FIGURE 5: artifact class.

The data class contains the Data object and Data store (fig. 6).

FIGURE 6: data class.

The concept of diagram is modelled thought the Business_Process class that acts as a container of pool.

3.2 Assertion About Classes

Each class has own assertions that represent the class features. In details, the class definition is created making intersections, unions and relationships between class and subClass. The used logical operators are:

- Or: this operator is used in the union statement and allows to define a new class as (inclusive) disjunction of several class;
- Not: It is used to define a new class as negation of another class;
- For instance, the disjoined class graphical_element is defined as:
- graphical_element EquivalentTo flow_object or (artifact or (connecting_object or (data or swimlane)));
- graphical_element SubClassOf BPMN_element;
- graphical_element SubClassOf not (supporting_element).

The following figure shows how the classes of meta-model was defined. The figure uses the set theory notation to describe the mutual relationship between class.

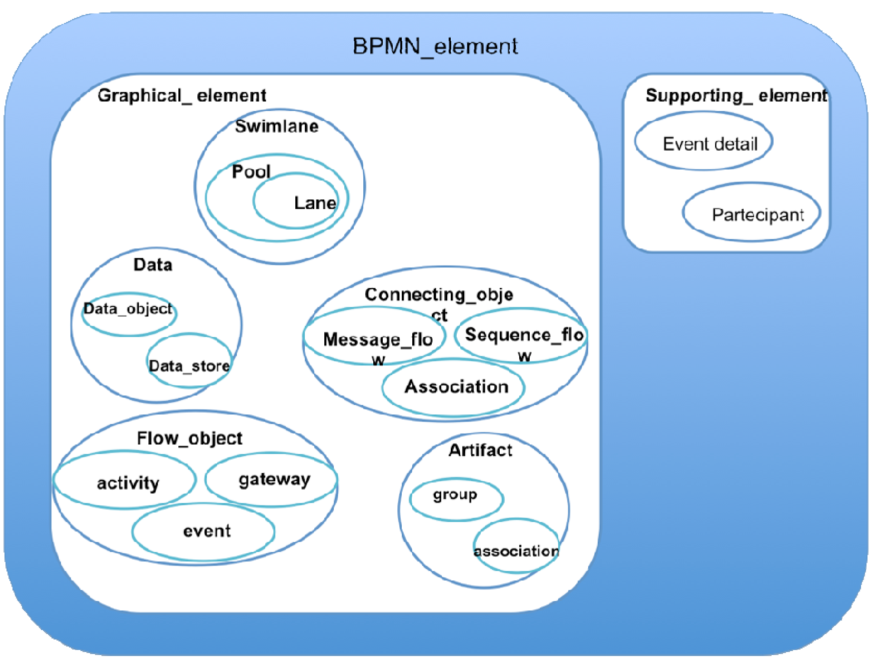

FIGURE 7: Existential class of the BPMN_element.

3.3 Defining the Class Through Class Property Restriction

To model the several types of graphical elements of BPMN 2.0, the event class, task class and gateway class are defined as restrictions on the properties of super-classes. In other words, the definition use restrictions on the set of individuals of the class itself.

The used restriction is a property restriction called existential. This restriction is defined in OWL as a set of individuals that have a specific property with relation ("some" operator) with the elements of another class. For instance, the class end_event is defined as:

end_event EquivalentTo event and (has_event_type some ({END}))

The following table reports all the existential property restriction of the meta-model.

Classes	Existential restriction
end_event	EquivalentTo Event and (has_event_type some ({END}))
intermediate_event	EquivalentTo Event and (has_event_type some ({INTERMEDIATE}))
start_event	EquivalentTo event and (has_event_type some ({START}))
gateway_data_based_exclusive	EquivalentTo gateway and (gateway_has_gateway_type some ({GATEWAY_DATA_BASED_EXCLUSIVE}))
gateway_inclusive	EquivalentTo gateway and (gateway_has_gateway_type some ({GATEWAY_INCLUSIVE}))
gateway_parellel	EquivalentTo gateway and (gateway_has_gateway_type some ({GATEWAY_PARALLEL}))
event_message	EquivalentTo event_detail and (has_event_detail_type some ({EVENT_MESSAGE}))
event_normal	EquivalentTo event_detail and (has_event_detail_type some ({EVENT_NORMAL}))
event_rule	EquivalentTo event_detail and (has_event_detail_type some ({EVENT_RULE}))
event_terminate	EquivalentTo event_detail and (has_event_detail_type some ({EVENT_TERMINATE}))
event_timer	EquivalentTo event_detail and (has_event_detail_type some ({EVENT_TIMER}))

TABLE 1: Existential Property Restriction.

To define the existential property restriction in the meta-model, specific classes are created through the definition of the instance contained in the classes themselves.

For instance, the class event_type is defined as enumeration of the all elements that are contained in itself:

event_type EquivalentTo {END, INTERMEDIATE, START }

The same technique allows to define the following classes:

Classes	Enumeration
event_type	EquivalentTo {END, INTERMEDIATE, START}
event_detail_type	EquivalentTo {EVENT_MESSAGE, EVENT_NORMAL, EVENT_RULE, EVENT_TERMINATE, EVENT_TIMER}
gateway_type	EquivalentTo {GATEWAY_DATA_BASED_EXCLUSIVE, GATEWAY_INCLUSIVE, GATEWAY_PARALLEL}
task_type	EquivalentTo {TASK_MANUAL, TASK_RECEIVE, TASK_SERVICE, TASK_USER}

TABLE 2: C dlasses Defined As Enumeration.

3.4 Object Property Definition

At this point we analyze properties that link together classes (Object Properties) a part from already defined relationships of specialization and generalization. We summarize some of these defining both domain and range of each property.

- Business_Process_Diagram_has_BPMN_element: Object Property has as domain Business_Process class and link to this class all BPMN_element defined;
- Business_Processt_Diagram_has_pool: has as domain the class Business_Process and as range the class Pool;
- pool_has_lane: the domain of the property is the class Pool and the range in the class Lane. It is used to design a Pool that must have one Lane or more than one lane(property cardinality greater than zero);
- lane_has_BPMN_element: Property that has as a domain the class Lane and as range the class BPMN_element. This relation allow to link to the lane the BPMN primitives that are in the lane.

The resulting ontology is represented in fig. 8.

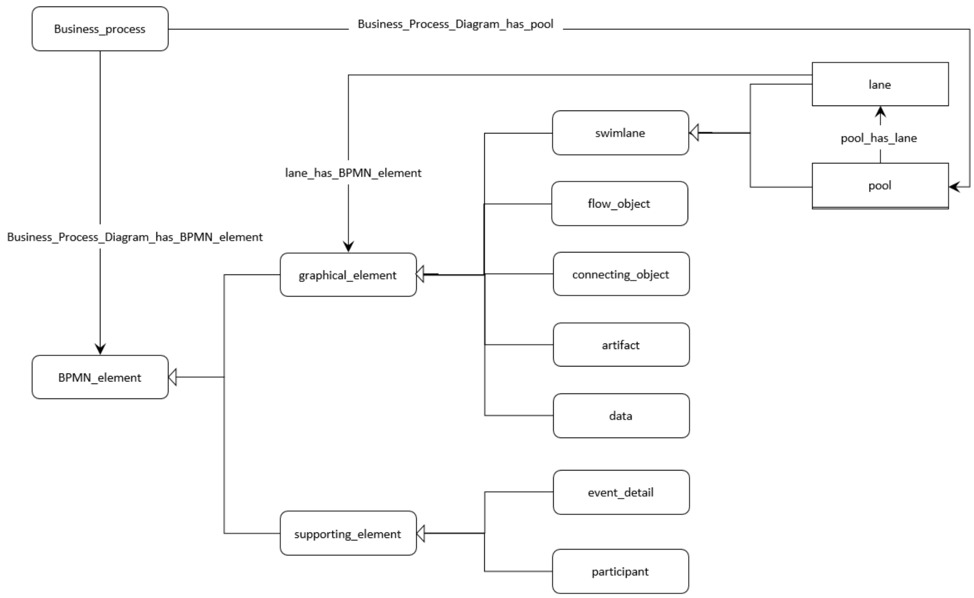

FIGURE 8: Representation of some BPMN meta-model Object Property.

In table 3 all properties of the meta-model are showed defining both domain and range.

4. IMPACT AND IMPROVEMENTS: A CASE STUDY

The added value of the research work is appreciated in the HSEPGEST project. The HSEPGEST (Management of Health, Safety, and Environmental Protection in enterprise processes) is a project funded from European Structural Funds 2007-2013.

In this project there was several knowledge base each for a specific aspect of the company. The knowledge base describes several aspects of the company such as company chart, law and regulation, goal, human resources, role and so on. Our contribution to the project was to introduce a tool for the management of the business process and to connect the business process to the knowledge already defined in the company. For example, in order to link a process lane to a well defined hierarchical position in a company chart, it is possible, having an ontological representation of the company chart, to link the two concepts (lane and position in a company chart).

We introduce the described BPMN ontological meta-model and a tool for the management of the business processes that allow not only to automate the business process but also to add a semantic layer to its description. Once saved the business process within a tool, it automatically become part of the company knowledge base and thus add new knowledge to it. The added value is that, because the business process is related to the knowledge base, a change to the knowledge base (for example a new regulation, a new position in the company chart and so on) trigger an alert to the designer: the designer know that a change was occurred in “something” in the company knowledge and what process is related to this change. The designer may change the business process design in order to answer to the company’s need.

It is clear that in this way the company knowledge is enriched by the business process that become part of the company’s knowledge. The same business process design, or part of it, may be used for further design in the company.

Another advantage comes from the possibility to use a business process template that is a part of the business process already defined from other goals but useful for the specific design problem. Using the business process template (annotated) in a new business process design, it is possible to use also the semantic annotation and thus enriching the overall business process design with these important features.

From these consideration it is clear that the impact and improvement introduced by the proposed research has been immediately tangible in a research project.

5. CONCLUSIONS AND FUTURE WORKS

In this paper we present an ontological meta-model for BPMN 2.0 notation. The meta-model is characterized by completeness (all primitives was considered) and simplicity (we define just the details useful for the design). The meta-model, written using OWL, is ready to accept a model of a specific business process defined using BPMN notation.

The uses of this meta-model are many: the meta-model can be used in a case tool for business process design in order to make semantic and syntactic consistency check. The meta-model can be used as knowledge base in order to link to the process semantic information that, otherwise, will be lost. In this way we introduce in the company’s knowledge base the concept of business process.

We think about the possibility to semantically enrich the process design with information not strictly related to the process flow but useful during the development of the application. These information can be added to the process design just creating an Object Property that allow to link the business process diagram to these information.

The future direction of our research work is to adopt Natural Language Processing techniques in order to extract from the company knowledge (for example law and regulation written in natural language) information useful to design business process. These information will be stored in a knowledge base and will be related to the business process design using semantic links.

Name of the relationship OBJECT_PROPERTY	**Domain**	**Range**
Business_Process_Diagram_has_BPMN_element	Business_Process	BPMN_element
Business_Process_Diagram_has_message_flow	Business_Process	message_flow
Business_Process_Diagram_has_has_pool	Business_Process	Pool
pool_defines_Business_process_diagram	pool	Business_Process
lane_has_BPMN_element	lane	BPMN_element
pool_has_lane	pool	Lane
pool_has_participant	pool	Lane
has_event_type	event	event_type
event_has_event_detail	event	event_detail
has_event_detail_type	event_detail	event_detail_type
has_event_start_trigger	start_event	event_message or event_normal or event_rule or event_timer
has_event_intermediate_target	intermediate_event	Activity
has_event_intermediate_trigger	Intermediate_event	event_message or event_normal or event_rule or event_timer
has_end_event_result	end_event	event_message or event_terminate
is_source_for_sequence_flow	activity or gateway or intermediate_event or start_event	sequence_flow
is_target_for_sequence_flow	Activity or gateway or intermediate_event or end_event	sequence_flow
is_source_for_message_flow	Activity or end_event or pool	message_flow
is_target_for_message_flow	activity or intermediate_event or pool or start_event	message_flow
is_source_for_association	artifact or data or flow_object	association
is_target_for_association	artifact or data or flow_object	Association
gateway_has_gateway_type	gateway	gateway_type
task_has_task_type	task	task_type
"has_link_to"	data_input data_output event_start intermediate_event gateway_data_based_exclusi ve gateway_inclusive gateway_parellel pool lane task	Knowledge base

TABLE 3: Object Properties' Domain and Range.

6. REFERENCES

[1] Object Management Group (OMG). (2011, January). Business Process Model and Notation (BPMN) Version 2.0. [On-line] 50. Available: http://www.omg.org/spec/BPMN/2.0/PDF [June 10, 2016].

[2] Object Management Group (OMG). (2006). Meta Object Facility (MOF) Core Specification. [On-line]. Available http://www.omg.org/spec/MOF/2.5 [June 10, 2016].

[3] H. M. Gholizadeh and M. A. Azgomi. "A meta-model based approach for definition of a multi-formalism modeling framework". *International Journal Computer Theory Engineering*, vol. 2, pp. 87-95, 2010.

[4] D. L. McGuinness and F. Van Harmelen. "OWL Web Ontology Language Overview," *W3C Recommendation*, vol. 10, pp. 1–22, 2004.

[5] R. Paiano, A. L. Guido, A. Pandurino. *Designing Complex Web Information Systems: Integrating Evolutionary Process Engineering: Integrating Evolutionary Process Engineering*. IGI Global, 2009 pp. 390.

[6] M. Hepp, F. Leymann, J. Domingue, A. Wahler, and D. Fensel, "Semantic business process management: a vision towards using semantic Web services for business process management" In *Proc. IEEE Int. Conf. E-bus. Eng.*,2005, no. 1, pp. 1–6.

[7] D. Martin, M. Burstein, and J. Hobbs, "OWL-S: Semantic markup for web services". Internet: https://www.w3.org/Submission/OWL-S/", Nov, 22, 2004 [June 10, 2016].

[8] M. Aslam, S. Auer, J. Shen, and M. Herrmann, "Expressing Business Process Models as OWL-S Ontologies," In *Business Process Management Workshops*, vol. 4103, 2006, pp. 400–415.

[9] W. Abramowicz, A. Filipowska, M. Kaczmarek, and T. Kaczmarek, "Semantically enhanced Business Process Modelling Notation," in *Proc. CEUR Workshop Proceedings*, 2007, pp. 1-4.

[10] C. Natschlager, "Towards a BPMN 2.0 ontology," in *Lecture Notes in Business Information Processing*, 2011, pp. 1–15.

[11] J. Prater, R. Mueller, and B. Beauregard, "An ontological approach to oracle BPM". *In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* 2012, pp. 402–410.

[12] M. Rospocher, C. Ghidini, and L. Serafini, "An ontology for the business process modelling notation," *in Frontiers in Artificial Intelligence and Applications,* 2014, vol. 267, pp. 133–146.

[13] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology,". Internet http://protege.stanford.edu/publications/ontology_development/ontology101.pdf,.2001. [June 10, 2016].