Data Transformations using QVT between Industrial Workflows and Business Models in BPMN2

Corina Abdelahad, Daniel Riesco
Departamento de Informática
Universidad Nacional de San Luis
San Luis, Argentina
{cabdelah, driesco}@unsl.edu.ar

Alessandro Carrara, Carlo Comin, Carlos Kavka
Research and Development Department
ESTECO SpA
Trieste, Italy
{carrara, comin, kavka}@esteco.com

Abstract— **Automation of business processes (workflows) has become today a key component to support the growth of organizations. Many standards in the field of business processes, however, are not directly applicable to the field of process engineering due to specific engineering workflow requirements. The Business Process Model and Notation (BPMN), a widely used notation to model business process which has been recently enhanced, allows to extend the use of workflows from the field of business process to the field of engineering. However, the large base of existing engineering workflows used currently by industry represents one of the main obstacles for the adoption of the new standards. Transformations play a pivotal role in the implementation of Model-Driven Architecture by providing a mechanism to express model refinement. Query/View/Transformation (QVT) is an OMG standard established to create queries, views and transformations of models. The QVT Relations, a component of QVT, allows to formalize model-driven transformations. ESTECO is a company which has developed a proprietary workflow modeler and an associated workflow engine. Even if its proprietary model has proven to be useful in the context of engineering processes, a standard-complaint industrial process flow will enable the building of unified and standardized models across all sectors of the business. This paper presents a model-to-model partial transformation using QVT between the ESTECO metamodel and the BPMN2 metamodel, where input data of ESTECO generates DataInputs, InputSets and IoSpecification for use in BPMN2. This transformation allows the conversion of most ESTECO industrial workflow to BPMN2, assuring portability between tools that support the BPMN2 standard.**

Keywords - BPMN2; business workflow; ESTECO; industrial workflow; metamodel.

I. INTRODUCTION

An industrial workflow is an automated business process usually used to execute complex processing tasks that requires many features that most business process models do not currently support [7]. These kinds of workflows are widely used in natural science, computational simulations, chemistry, medicine, environmental sciences, engineering, geology, astronomy, automotive industry and aerospace among other fields.

BPMI (Business Process Modeling Initiative) together with the OMG have developed the widely used BPMN notation for modeling business processes [2][5]. BPMN defines a formal notation for developing platform-independent business processes, opposed to specific definitions of business processes such as XPDL (XML Process Definition Language) [9] or BPEL4WS (Business Process Execution Language for Services Web) [8]. BPMN defines an abstract representation for the specification of executable business processes within a company, which can include human intervention, or not. BPMN also allows collaboration between business processes of different organizations. The last definition of the BPMN standard (the release 2.0) has been developed by taking as one of its objectives the overcoming of the limitations that prevented its use in scientific and engineering applications [1][2].

The definition of this new standard allows, for the very first time, to extend the use of workflows from the field of business process to of the field of engineering. Engineering workflows, which share many properties with well-known scientific workflows, are heavily used in industry today. Although they are widely used, there is currently no standard accepted for the definition of engineering workflows, despite the efforts of standard organizations in the field of business processes. The large base of existing engineering workflows used currently by industry, which will need to be transformed between proprietary legacy formats to the new standard in order to be executed, represents one of the main obstacles for the adoption of the new standards.

QVT is a standard relation language for model transformation defined by the OMG with a specification based on MOF and OCL[17]. The language consents to express a declarative specification of the relationships between MOF models and metamodels supporting complex object pattern matching. A QVT transformation defines the rules by which a set of models can be transformed into a different set [4]. Furthermore, it specifies a set of relations that the elements of the implicated models in the transformation must fulfill. The model types are represented by their corresponding metamodels. A relation in QVT specification consists in a set of transformation rules where a rule contains a source domain and a target domain [6]. A domain is a set of variables to be matched in a typed model, with each domain defining a candidate model and also having its own set of patterns [4].

 ISBN: 978-1-61208-230-1

The paper is structured in sections. Section II presents related works while section III describes the motivations of current research. The ESTECO metamodel used as the source model for transformations is described in section IV. Sections V and VI present the transformation architecture and the transformation between models respectively. The paper ends with conclusions in section VII.

II. RELATED WORK

Several works in the field of software engineering are related to the concept of transformation between models, and many of them use BPMN to model business process. To the best of our knowledge, no other research work has considered BPMN2 as the target model for transformation in the context of industrial workflows.

Marcel van Amstel et al. [12], investigate what factors have an impact on the execution performance of model transformation. This research estimates the performance of a transformation and allows to choose among alternative implementations to obtain the best performance.

In this same line, a model-to-model transformation between PICTURE and BPMN2 is presented in [11]. PICTURE is a domain-specific process modeling language for the public administration sector. The transformation allows to model administrative processes in PICTURE and to get BPMN2 models for these processes automatically, helping electronic government by making possible the implementation of supporting processes.

In [13], three sets of QVT relations are presented as a mean of implementing transformations in a model-driven method for web development. One of them transforms a high-level input model to an abstract web-specific model. The other two transform the abstract web model to specific web platform models.

An example application is presented in [14] to demonstrate an automated transformation of a business process model into a parameterized performance model.

III. MOTIVATIONS

As mentioned before, engineering workflows are heavily used in industry today to execute complex processing tasks like simulation or optimization [3]. Current examples include the areas of automotive, aerospace, turbines and other industries where the development of complex products is modeled as an engineering process defined in terms of the collaboration of various engineering services with usually large exchange of information between them.

Both scientific and engineering workflows differ from business workflows in many aspects. For example, business workflows usually deal with discrete transactions, but engineering and scientific workflows in most cases deal with many interconnected software tools, large quantities of data with multiple data sources and in multiple formats. Also, engineering services have usually a very long execution duration and depend on the execution environment.

Even if engineering workflows have been used successfully since many years, most of the tools used to define and execute them are not based on standard technologies. Until now, a single standard could not be used to represent both the abstract view (used by the engineer to represent the process at the scientific domain) and the workflow representation used for execution (at workflow engine level). The use of standards like BPMN for abstract representation and BPEL for execution were proposed in the past, but never went too far in industry due to the need to support two different standards for the same workflow. BPMN2, however, defines a standard with support for both levels, with many different graphic editors and workflow engines available, making a business standard accessible for the very first time to industry to completely support engineering workflows.

With BPMN2, many companies will be tempted to support a standard workflow for engineering applications. However, it must be considered that there exists a large base of engineering workflows already designed and used currently by industry which cannot be just thrown away. In order to provide legacy workflow support, we propose a methodology for the transformation of legacy proprietary workflows into BPMN2 standard workflows. This approach will provide an extra incentive for companies to abandon proprietary workflows and move to standard technologies coming from the field of business processes.

However, the transformation is not without pain. The extra data and process requirements in engineering workflows need to be handled properly. Fortunately, BPMN2 has been defined with an extension facility which allows to add required constructions without breaking standard compliance.

As part of the methodology, this work presents a partial transformation for selected constructions of a widely used industrial engineering workflow to BPMN2 in order to present a valid path to perform legacy workflow conversion to a well-defined standard. The next section presents a short introduction to the legacy metamodel.

IV. ESTECO METAMODEL

The metamodel selected as an example is the workflow model used by ESTECO for modeling simulation workflows in the context of industrial multi-objective optimization. This workflow, which is typical in this kind of environments, includes one task node for each activity and data nodes used to represent input, output and temporary data objects. Data objects can represent simple data like integer, doubles, vectors, matrix or more complex data like files or databases. Activities correspond to the execution of simulators, scripts and other applications in local or remote locations. Usually each activity has associated a set of configuration files, which can be large (many gigabytes being common), and a set of inputs and outputs (which can also be very large files or databases). Distributed execution is required, meaning that the activities specified in the workflow can be executed in different nodes (on the grid or the cloud system), requiring data to be passed between them.

The next section provides a description of the framework used for the transformation by applying it to a subset of ESTECO's workflow.

 ISBN: 978-1-61208-230-1

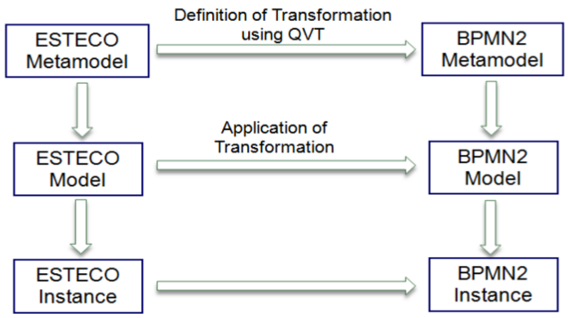

Figure 1. Transformation architecture.

V. TRANSFORMATION ARCHITECTURE

Our proposal aims to apply the most recent concepts of business processes to the field of engineering workflows in industrial fields. The use of standards in industry is important since it guarantees portability between tools that support BPMN2.

The industrial legacy workflow selected has an XML representation, allowing the use of tools like QVT for transformation. There is no one-to-one correspondence between the different components of ESTECO's workflow and BPMN2 constructions, since control nodes and data nodes are very differently handled in both models. Also, files and database handling put extra requirements which can only be handled properly with BPMN2 extensions.

The QVT transformations describe relations between the source metamodel and the target metamodel, both specified in MOF. The transformation defined is then applied to a source model, which is a part of ESTECO source metamodel, to obtain a target model, which is part of the BPMN2 target metamodel, as can be seen in Fig. 1.

The metamodels used in the definition of the transformation are shown at the top level. The specific models to which the transformation defined in the metamodel level is applied in order to obtain BPMN2 models is shown at the middle level. The lower level represents the instances of the models which can be executed in the corresponding workflow engines.

As mentioned before, activities and processes need data in order to be executed, and in addition, they can produce data during or as a result of their execution. In BPMN2, data requirements are captured as *DataInputs* and *InputSets*. The produced data is captured using *DataOutputs* and *OutputSets*. These elements are aggregated in an *InputOutputSpecification* class [2] as can be seen from Fig. 2. The *DataInputs* and *DataOutputs* are additional attributes of the *InputOutputSpecification* element; these elements are optional references to the *DataInputs* and *DataOutputs* respectively. A *DataInput* is a declaration that a particular kind of data will be used as input of the *InputOutputSpecification*. A *DataOutput* is a declaration that a particular kind of data can be produced as output of the *InputOutputSpecification*. *DataInputs* and *DataOutputs* are *ItemAware* elements. If the *InputOutputSpecification* defines no *DataInput*, it means no data is required to start an Activity. If the *InputOutputSpecification* defines no *DataOutput*, it means no data is required to finish an Activity [2].

A partial view of the ESTECO metamodel with the metaclasses involved in the relations described in this work is shown in Fig. 3. The *TInputDataNode* and

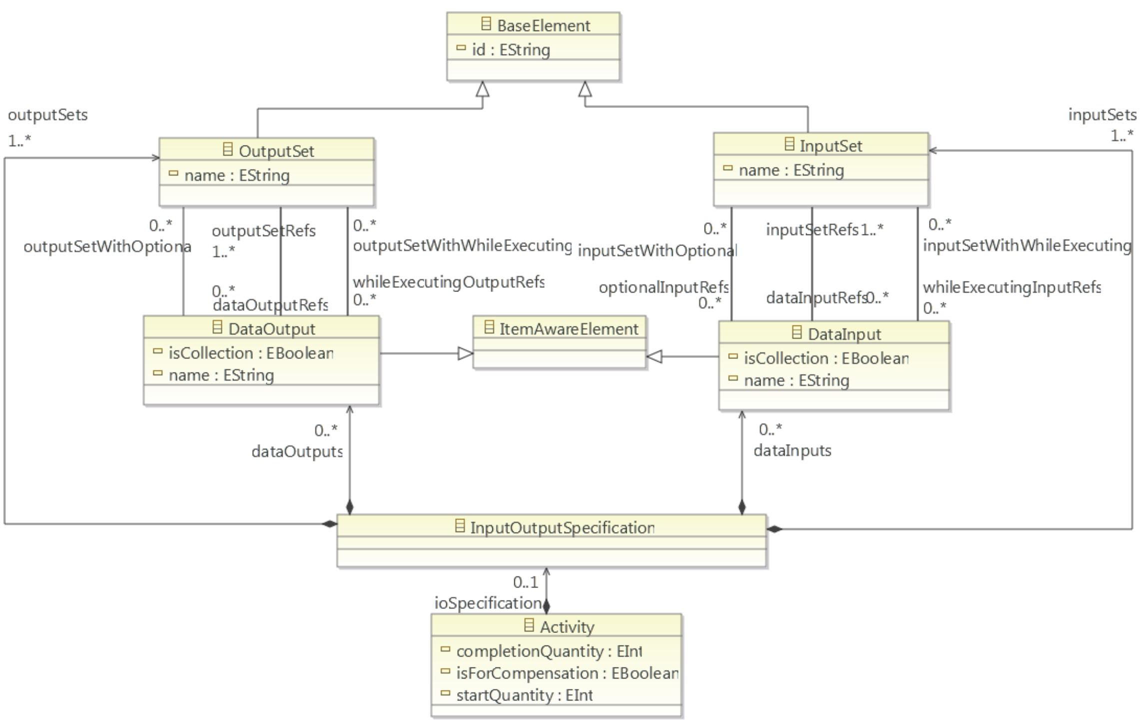

Figure 2. Partial view of the BPMN2 metamodel.

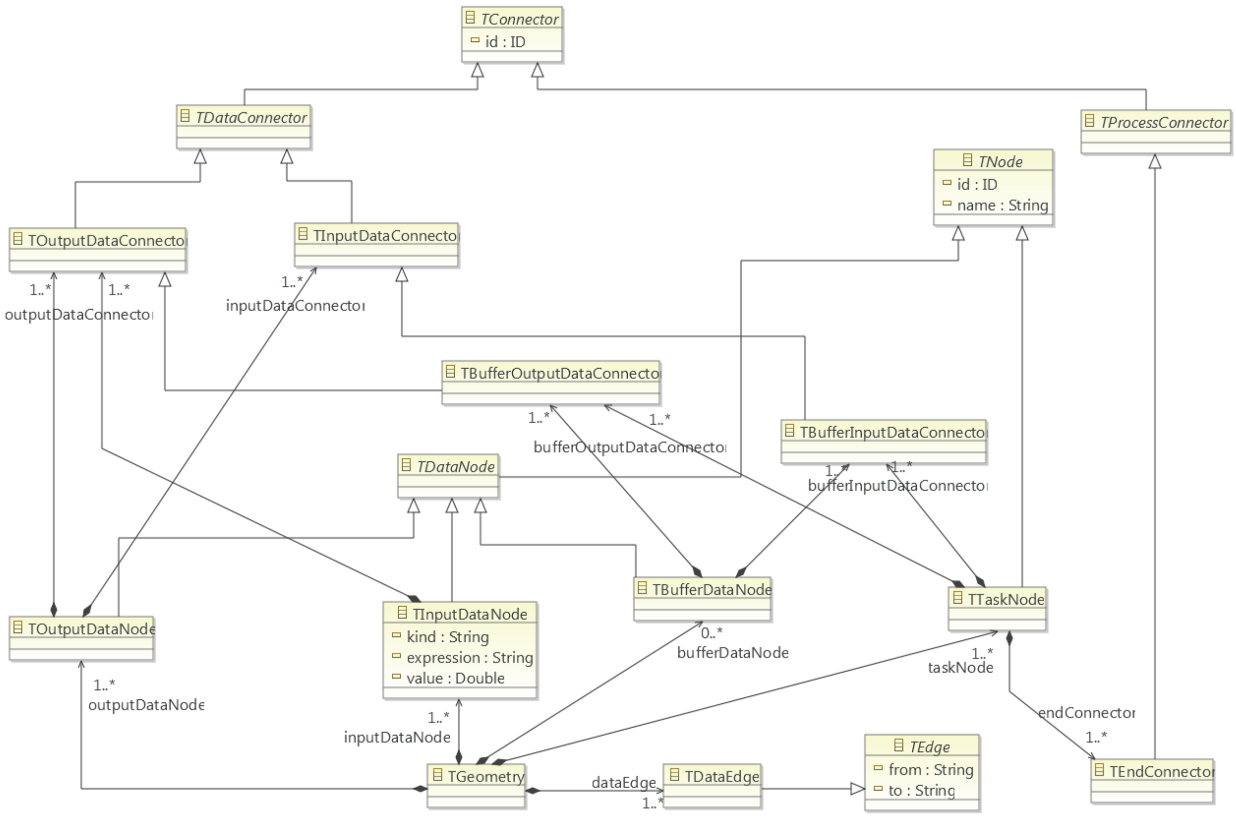

Figure 3. Partial view of ESTECO metamodel.

TOutputDataNode elements inherit the attributes and model associations of *TDataNode*, which in turn, inherits from *TNode*. The *TGeometry* class is the outermost object for all ESTECO elements, i.e., all these elements are contained in a *TGeometry*. The *TInputDataNode* element is a particular kind of *TDataNode* that will be used as input of *TGeometry* to a *Task*. The *TOutputDataNode* element is a particular kind of *TDataNode* which can be produced as output of a *Task* contained in *TGeometry*. A *TTaskNode* class represents the task that is performed within an industrial workflow.

VI. TRANSFORMATION BETWEEN MODELS

A transformation specifies a group of relations that the elements of the involved models must fulfill. A transformation may have any number of input or output parameters known as domains. For each output parameter, a new model instance is created according to the metamodel of the output metamodel (in this case, the metamodel BPMN2).

Each domain identifies a corresponding set of elements defined by means of patterns. A domain pattern can be considered an object template. A relation in QVT defines the transformation rules. A relation implies the existence of classes for each one of its domains. In a relation, a domain is a type that may be the root of a template pattern. A domain implies the existence of a property of the same type in a class. A pattern can be viewed as a set of variables and a set of constraints that model elements must satisfy. A template pattern is a combination of a literal that can match against instances of a class and values for its properties. A domain can be marked as *checkonly* or *enforced*. A *checkonly* domain simply verifies if the model contains a valid correspondence that satisfies the relation. When a domain is *enforced*, if checking fails, the elements of target model can be created, deleted or modified so as to satisfy the relationship.

A relation can contain two sets of predicates identified by a *when* or a *where* clause. The *when* clause specifies the condition that must be satisfied to execute the transformation. The *where* clause specifies the condition that must be satisfied by all model elements involved in the relation, and it may contain any variable involved in the relation and its domains [4]. In the context of transformation, a model type represents the type of the model. A model type is defined by a metamodel and an optional set of constraint expressions.

The transformation between ESTECO metamodel and BPMN2 metamodel is defined as follows:

```
transformation ESTECOToBPMN2(source : esteco_m,
                             target : bpmn2)
```

This transformation takes as input an ESTECO model, which is an instance of the ESTECO metamodel, and produces a BPMN2 model, that will be an instance of the BPMN2 metamodel.

Below, the relations which define the mapping between ESTECO metamodel classes and BPMN2 metamodel classes are shown. This correspondence is not straightforward. As we mentioned in the previous section, the *DataInput*s are captured in *InputSet*s and both are added into an *InputOutputSpecification*.

The same happens with the *DataOutput*s. So, in the transformation it is necessary to generate an *IoSpecification* object to aggregate *DataInput*s, *DataOutput*s, *InputSet*s and *OutputSet*s.

The relation used to create an *IoSpecification* object is shown below:

```
relation createIOSpecificationTask {
   checkonly domain source g:esteco_m::TGeometry  { };
   enforce domain target t:bpmn2::Task  {
   ioSpecification= ioSpecif :
                              bpmn2::InputOutputSpecification {}
  };
  primitive domain id_task:String;
  where {
   getDataInputTask(g,ioSpecif, id_task);
   createInputSetsTask(ioSpecif,ioSpecif);
   getDataOutputTask(g, ioSpecif, id_task);
   createOutputSetsTask(ioSpecif, ioSpecif);
   }
}
```

The relations used to create *InputSet*s and *OutputSet*s is presented below.

```
relation createInputSetsTask {
  checkonly domain target ioSpecif:
                             bpmn2::InputOutputSpecification {
  };
  enforce domain target ioSpecif :
                             bpmn2::InputOutputSpecification {
        inputSets = input_set :bpmn2::InputSet{
        dataInputRefs= ioSpecif.dataInputs
      }
  };
}
relation createOutputSetsTask {
  checkonly domain target ioSpecif:
                             bpmn2::InputOutputSpecification{
  };
  enforce domain target ioSpecif :
                             bpmn2::InputOutputSpecification{
       outputSets = output_set :bpmn2::OutputSet{
       dataOutputRefs= ioSpecif.dataOutputs
      }
   };
}
```

Note that an *InputSet* is a collection of *DataInput* elements that together define a valid set of data inputs associated to an *InputOutputSpecification*. An *InputOutputSpecification* must define at least one *InputSet* element. An *OutputSet* is a collection of *DataOutput*s elements that together can be produced as output from an Activity. An InputOutputSpecification element must have at least OutputSet element [2].

The relations used to obtain the *DataInputs* of the ESTECO model and the generation of *DataInputs* in BPMN2 is presented below.

```
relation getDataInputTask{
   id_input, name_input : String;
   value_input : Real;
   checkonly domain source g:esteco_m::TGeometry{
     taskNode = t:esteco_m::TTaskNode{
     bufferInputDataConnector = buffer_input :
                         esteco_m::TBufferInputDataConnector {}
     },
     inputDataNode = input : esteco_m::TInputDataNode {
       id = id_input,
       name = name_input,
       value = value_input,
       outputDataConnector = output_data :
                         esteco_m::TOutputDataConnector {}
     },
     dataEdge = data_edge : esteco_m::TDataEdge {}
   };
   enforce domain target ioSpecif :
                             bpmn2::InputOutputSpecification  {
         dataInputs = data_input : bpmn2::DataInput {
         id= id_input + '_T',
         name = name_input,
         itemSubjectRef = item : bpmn2::ItemDefinition {
             id = 'DoubleItemDefinition'
         }
       }
   };
   primitive domain id_task:String;
   when {
     if (data_edge.from = output_data.id) and
       (data_edge.to = buffer_input.id ) and
       (id_task=t.id) then true else false
  endif;
  }
}
```

Each data input of ESTECO must be transformed into a data input of BPMN2. This transformation is straightforward; the QVT code presented before shows the procedure by which the *id*, *name*, *value* and *connectors* are obtained.

The relation used to obtain the *DataOutput*s of ESTECO model and the generation of *DataOutput*s in BPMN2 is shown below.

```
relation getDataOutputTask{
   id_output, name_output : String;
   checkonly domain source g:esteco_m::TGeometry {
      taskNode = t:esteco_m::TTaskNode{
      bufferOutputDataConnector = buffer_output :
                    esteco_m::TBufferOutputDataConnector {}
      },
      outputDataNode = output :
                              esteco_m::TOutputDataNode {
         id = id_output, name = name_output,
         inputDataConnector = input_data :
                              esteco_m::TInputDataConnector {}
      },
      dataEdge = data_edge : esteco_m::TDataEdge {}
   };
   enforce domain target ioSpecif :
                           bpmn2::InputOutputSpecification  {
      dataOutputs = data_output : bpmn2::DataOutput {
      id= id_output + '_T',
      name = name_output,
      itemSubjectRef = item : bpmn2::ItemDefinition {
         id = 'DoubleItemDefinition'      }
      }
   };
   primitive domain id_task:String;
   when {
      if (data_edge.from = buffer_output.id) and
         (data_edge.to = input_data.id )  and
         (id_task=t.id) then  true else false
      endif;
   }
}
```

VII. CONCLUSION

In last years, business processes have gained popularity and have been subject to a large number of studies.

In the context of engineering, the execution of many parallel activities with complex interdependencies is required. At the same time, configuration control of the models should be maintained in order to ensure the traceability of the experiments, a requirement that is not necessarily considered in the typical business models. The efficient integration with platforms such Service Oriented Architecture (SOA) and Cloud Computing is also essential in the context of industrial workflows, a feature that is not considered in typical business workflows [15][16].

The objective of this work has been to apply the latest concepts of business processes to the industrial field. Furthermore, it intended to show the importance of the use of standards in the industrial field to guarantee portability between tools that support BPMN2.

In order to validate experimentally the methodology, the approach has been applied to the engineering environment supported by a company specialized in multi-objective optimization. Even if the company is currently working to fully support the standard for future workflows, the approach presented in this paper will allow to guarantee the support for legacy workflows by performing a transformation between the old metamodel to the BPMN2 standard metamodel. It is important to stress that this transformation allows the conversion of most ESTECO industrial workflows to BPMN2 consenting their execution in BPMN2 workflow engines with adequate extensions support.

ACKNOWLEDGMENT

The authors thank the reviewers for the very useful comments that have contributed to enhance the paper.

REFERENCES

[1] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox, Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers, "Examining the challenges of scientific workflows". IEEE Computer vol. 40, no. 12, 2007, pp. 24-32.

[2] Object Management Group (OMG), "Business process model and notation", http://www.omg.org/spec/BPMN/2.0 [retrieved: October, 2012]

[3] ESTECO S.p.A., "modeFRONTIER applications across industrial sectors involving advanced CAD/CAE packages", http://www.esteco.com/home/mode_frontier/by_industry, [retrieved: October, 2012]

[4] Object Management Group (OMG), "Meta object facility (MOF) 2.0 query/view/transformation, V1.1", http://www.omg.org/spec/QVT/1.1 [retrieved: October, 2012]

[5] The Business Process Management Initiative (BPMI.org), http://www.bpmi.org/ [retrieved: October, 2012]

[6] Li Dan, "QVT based model transformation from sequence diagram to CSP", 15th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS), 2010, pp. 349-354.

[7] Li Hongbiao, Li Feng, and Yu Wanjun, "The research of scientific workflow engine", IEEE International Conference on Software Engineering and Service Sciences (ICSESS), 2010, pp. 412-414.

[8] Oasis, "Web services business process execution language version 2.0", http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html, [retrieved: October, 2012]

[9] Workflow Management Coallition (WfMC), "XML Process Definition Language", http://www.xpdl.org, [retrieved: October 2012]

[10] Narayan Debnath, Fabio Zorzan, German Montejano, and Daniel Riesco. "Transformation of BPMN subprocesses based in SPEM using QVT", IEEE International Conference on Electro/Information Technology, 2007, pp. 146-15.

[11] Henning Heitkoetter, "Transforming PICTURE to BPMN 2.0 as part of the model-driven development of electronic government systems", 44th Hawaii International Conference on System Sciences (HICSS), 2011, pp. 1-10.

[12] Marcel van Amstel, Steven Bosems, Ivan Kurtev, and Luís Ferreira Pires, "Performance in model transformations:

experiments with ATL and QVT", Lecture Notes in Computer Science, Volume 6707, Theory and Practice of Model Transformations, Springer, 2011, pp. 198-212.

[13] Ali Fatolahi, Stéphane Somé, and TimothyLethbridge, "Automated generation of abstract web models using QVT relations", Technical Report TR-2010-06, School of Information Technology and Engineering, University of Ottawa, September 2010.

[14] Paolo Bocciarelli and Andrea D'Ambrogio, "A BPMN extension for modeling non functional properties of business processes", TMS-DEVS'11 Proceedings of the 2011 Symposium on Theory of Modeling & Simulation: DEVS Integrative M&S Symposium, Springer-Verlag, 2011, pp. 160-168.

[15] Cui Lin, Shiyong Lu, Xubo Fei, Artem Chebotko, Darshan Pai, Zhaoqiang Lai, Farshad Fotouhi, and Jing Hua, "A reference architecture for scientific workflow management systems and the VIEW SOA solution", IEEE Transactions on Service Computing, vol. 2, no. 1, 2009, pp. 79-92.

[16] Gideon Juve and Ewa Deelman, Scientific workflows and clouds, ACM Crossroads, vol. 16, no. 3, 2010, pp. 14-18.

[17] Object Management Group (OMG), "Modeling and metadata specifications", http://www.omg.org/spec, [retrieved: October 2012]