Developing a Business Application with BPM and MDE

H. Fernández-Fernández, E. Palacios-González, V. García-Díaz1, B. Cristina Pelayo G-Bustelo,
Óscar Sanjuán Martínez, Juan Manuel Cueva Lovelle,

Department of Computer Science, University of Oviedo, Oviedo, Asturias, Spain

Abstract – **In this paper we have designed an architecture for the generation of a business application, that allows to business users to adapt their processes to the constant change. At the moment all the architectures based to a great extent on SOA allow to modify the processes in a short period of time, but we go beyond and give the possibility to the business user of modifying their processes. To design this architecture, we rely on the fundamental use of two technologies: BPM (Business Process Modeling) and MDE (Model Driven Engineering). Inside these technologies we focus on the creation of a business process notation extended from BPMN that is agile, easy to learn and design, and capable to provide semantic information about the process. Therefore this notation allows business process to modify their processes to achieve the proposed goal.**

Keywords: Business Process Modeling Notation (BPMN), XML Process Definition Language (XPDL), Atlas Transformation Language (ATL), Model Driven Engineering (MDE).

I. INTRODUCTION

NOWADAYS the increasing tendency to the development of applications based in business processes have triggered a new need due to the constants changes well for restructuring of the organization or for the improvement of their processes that the applications may suffer. For this motive one of the most increasing requirements on the part of the users of business is the possibility of adjustment to the changes, through applications that allow them to manage their own business processes.

The goal of this paper is to present an architecture to offer this functionality which is required by the users. In the practice we have applied this architecture to an application about the food traceability of *"Cabrales Cheese"* [1], but we would apply it to any other application. Initially we begin from the development of a MDE [2], actually a MDA (Model Driven Architecture) [2], which generate an Web Application in ASP.NET. This paper is structured as follows: Section 2 similar architectures at the moment. Section 3 designed architecture for business application. Section 4 generation of the application. Section 5 conclusion. Section 6 future work.

II. SIMILAR ARCHITECTURES AT THE MOMENT

At the moment we have not found architectures for the design of applications that allows the business user to modify or to add new processes. Therefore we will present those architectures that, being based on BPM and MDA, they try to orientate the user towards the development of applications that satisfy their goals. Previously we need to define what we understand for BPM and MDA.

For BPM we understand those graphical notations that allow us to represent the business processes of a certain organization or company.

For MDA [1,3,4] we understand those architectures that driven by models try to separate the functional specification of the system to the specification of the implementation of this functionality in a specific platform.

Once clarified the area of the technologies that allow us to filter the existing architectures, we will continue to see them in detail.

A. BPM with Activity Diagrams UML 2.0 and MDA

This architecture is characterized for being easy and simple of integrating due to the great relation that exists between the activity diagrams and the class diagrams. The class diagrams will allow to MDA to generate the final application.

1) Stage of description

This architecture focuses on the use of activity diagrams for the representation of the business processes. In the work [5] the activity diagrams are obtained and through of transformations based on code the class diagrams are achieved. Each activity of the activity diagram is mapped to one class in the class diagram. Once obtained the class diagrams they are moved to a MDA tool that will generate the specified application from the class diagrams.

In the work [6] we pretend to generate Web services from the activity diagram. Thank to an UML profile these diagrams provide enough information to be transformed to class diagrams and finally, through the class diagrams, MDA will generate the Web services.

2)   Stage of review

These architectures provide various easy and simple transformations of the activity diagrams to the class diagrams, but they were rejected as a point of departure for our architecture due to the lack of expressiveness that has the activity diagrams UML 2.0 with regard to another notation for the modelled of business process as BPMN (Business Process Modeling Notation) [7]. This lack of expressiveness it is commented in detail in [8,9].

B. BPM with BPMN and MDA

This architecture is characterized for being one of most used and complex, moreover in the most of cases it is completely orientated to services.

1)   Stage of description

This architecture focuses on BPMN use on the level CIM of the MDA. BPMN is a standard developed and promoted by BPMI (Business Process Management Initiative), whose principal goal is to provide an understandable notation for anyone, from analysts to business users as well as to assure that the languages for the business process execution could be visualized by a common notation.

In the works [10,11], applications orientated to SOA are generated. Therefore BPMN diagrams are transformed to a process model executable language, which in this case is WS-BPEL (Web Service Business Process Execution Language) [12, 13, 14]. WS-BPEL is a standard defined by OASIS (Organization for the Advancement of Structured Information Standards), capable of specify, to achieve the automation of the business processes, to orchestrate the multiple activities of the Web services, to interpret and to execute the processes following a certain architecture. Once obtained the model WS-BPEL it will be generated the whole application with MDA help.

In the works [15, 16] is proposed to pass from BPMN to UML 2.0 activity diagrams and later, from these, pass to class diagrams or to class diagrams directly as it happens in [15], where transformations are made through QVT (Query/View/Transformation)[17]. In the work [16] is considered to pass from BPMN to class diagrams with the addition of an ontological search of terms belonging to the domain of the application, which provides help to identify classes of the domain, to be able to come to a class diagrams checked enough. Once obtained the class diagrams, MDA will generate the application.

2)   Stage of review

In the works [15, 16] even using BPMN as notation for the business process modelled, an information loss is produced during the pass from BPMN to class diagrams, because they are not capable of express clearly error notations and exceptions and some others aspects.

On the other hand, the works [10,11] this information loss is not produced in the transformation from BPMN to WS-

BPEL. Moreover the transformation from BPMN to WS-BPEL is the most supported by all the both commercial and free tools existing nowadays (Intalio, Oracle BPEL Process Manager, WebMethods) so this transformation is practically direct. The proposed architectures in the works [10,11] were considered as a point of departure for ours.

C. BPM with Owner Business Process Modeling Notations and MDA

This architecture is characterized for being most complex and less used at the moment.

1)   Stage of description

This architecture in the work [18] presents the integration between two commercial tools: Bizzdesigner and OptimalJ. This integration tries to combine the design and analysis of business process in enterprises application development based in MDA.   Bizzdesigner [19] is used for the design, analysis, documentation and information related with business processes. Bizzdesigner uses one owner notations for BPM.

OptimalJ [20] is a MDA implementation based in Eclipse [21]. OptimalJ allows a quick design, development and deployment of J2EE applications. This architecture uses Bizzdesigner to design and model the business process on the level CIM, whereas for the level PIM and PSM use OptimalJ.

2)   Stage of review

This architecture quickly is discarded because use one owner notation for the Business Process Modeling. This is a problem which involves explaining to business users this notation moreover tool dependence on Bizzdesigner. Therefore this does not allow using another tool or notation for the Business Process Modeling to business users.

III.   DESIGNED ARCHITECTURE FOR BUSINESS APPLICATION

The designed architecture has as goal to allow to business users adjustment to the constant changes that suffer their processes. For this reason it allows to any user himself to modify the business processes, in order to adapt them to the new needs that are produced constantly. These modifications will produce changes in the behavior of the application through the architecture proposed by OMG for MDA.

We applied this architecture to our application of food traceability of *"Cabrales Cheese",* beginning from an application based on MDA, already implemented and into operation with its CIM, PIM and specified PSM. Below we show the business process model for the application *"Cabrales Cheese".*
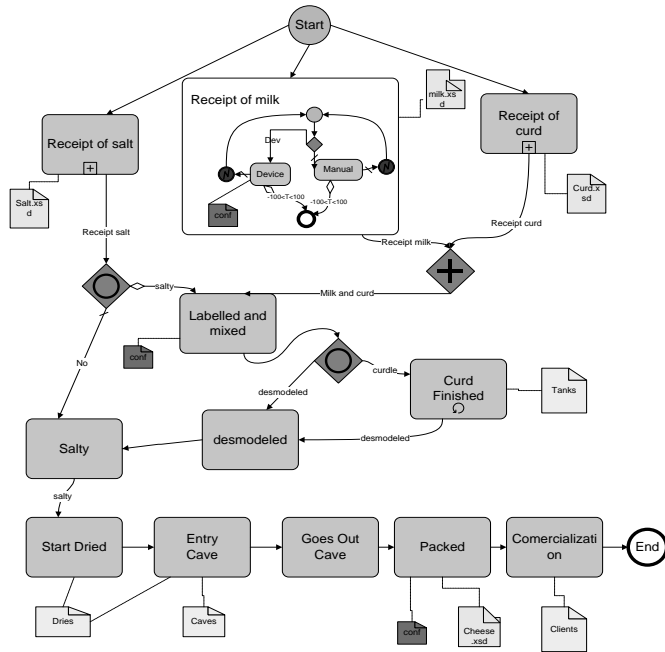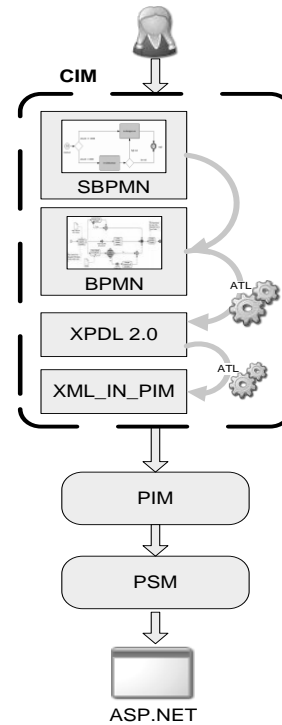
Fig. 2 BPMN – Application *Cabrales Cheese*

Fig. 3 Designed Architecture

The principal difference of this architecture is going to has one more layer of abstraction belonging to the level CIM. This layer will include the graphical notation that allows representing the business processes of the application.

This graphical notation is BPMN which is the one more extended nowadays for business processes. BPMN is a notation understood by analysts, developers and business users but the last group of users are not capable of use such notation to modify its business processes without need any technical knowledge [22]. To solve this problem we propose an extended notation of the own BPMN but easier, agiler and simpler for the business users, so that it allows them to modify the business processes without any need of technical knowledge. We show below the designed architecture with the levels proposed by MDA. In this figure we focused more the level CIM which is most important.

Inside of the level CIM one important layer is the referred to the transformations between the business process diagrams and the model PIM of our MDA, in which we make two transformations with ATL [23]. One of them model – model of BPMN to XPDL [24] and the other one mode – text of XPDL to PIM.

XPDL (XML Process Definition Language) developed by WfMC (Workflow Management Coalition) [24] is a language both textual and graphical that allows us to model the business processes but focus ourselves more on how it may be the work-flow, in order to achieve business goals. One of the most important XPDL advantages is the interoperability that offers between all the tools that support it, as far as it is a XML file that represents even the coordinates X and Y of all the elements that need graphical representation. We use XPDL 2.0 because its version 1.0 as it is said in [25,26] has little expressiveness and does not support all the BPMN elements.

Once done the transformations BPMN to XPDL and XPDL to PIM, we obtain the input XML file which is send to the PIM. In this moment we will already have everything which is necessary for the application independently of the platform in an only XML file. Therefore we can already generate our application. Principally we have to comment that our application is not orientated to services due to requirements of the user, what means an added difficulty with regard to the architectures taken as point of departure.

We follow the paper on those points that involve an added difficulty at the moment of apply our architecture.

A. The Simple Business Process Modeling Notation - SBPMN

As it is commented in [22], nowadays BPMN is a notation understood by analysts and business users but never a business user is able to make a diagram with BPMN himself, due it contains terms and properties with technical character. For this reason it is proposed the use of SBPMN (Simple Business Process Manager Notation) that follows the standard defined by BPMN excluding and changing those concepts that could need technical knowledge. With this goal we develop an editor for SBPMN Fig. 3. This editor has as an objective to provide major simplicity and to help the business user at the moment of creating a SBPMN diagram. Someone of the points most distinguished of this editor are:

- Contextual helps for every notation element, allowing the user to know all the possibilities offered by each element.

- Control of errors, indicating the possible solutions that the user should take to solve the error.

- Validation of the diagrams, verifying that the represented diagram fits the notation SBPMN and certainly BPMN.
- Reduces the decision tasks on the part of the of business user at the moment of selecting one element or another.
- Exportation of the diagram to XPDL 2.0 format, allowing to extend this functionality to include any other format as WS-BPEL.
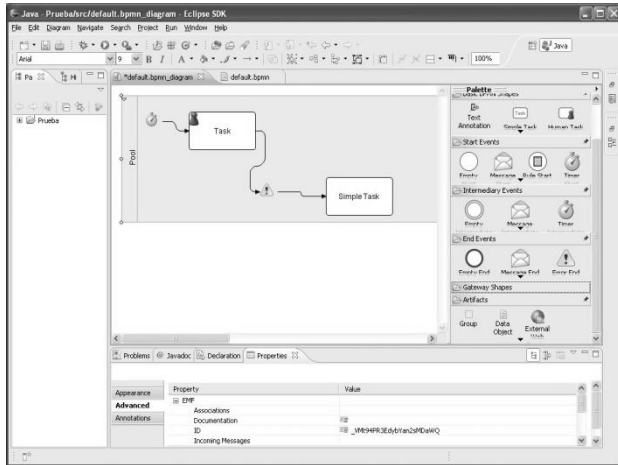
Later we show a picture of the editor SBPMN.

Fig. 4 Editor SBPMN

One of the points that reduces the decision tasks in the business user's side, is the use of gateways or tasks elements.

Among the main elements of the notation we make a reform in the offered tasks in BPMN. Actually BPMN only offers one element, Simple Task. In SBPMN we classify the tasks in three types: Human Task, Automatic Task and Simple Task, these task provide us more semantic information about the type of the process. This type of additional information had been really important for the authors of the work [10] where they created a group of primitives in the business process modeling for the creation of a navigability diagram through BPMN. Next we show in the Table 1 the different type of tasks in SBPMN.

BPMN	SBPMN
Simple Task	Simple Task
	Human Task
	Automatic Task

Table 1 BPMN Task and SBPMN Tasks

In BPMN the user is about to choose the gateway that fits his problem, on the other hand in SBPMN the user choose an unique gateway and, it depends on the needs required by it, the user will introduce some kind of parameters or other ones in order to define the functionality of it. Later with these parameters as a point of departure, the editor will transform in the exportation process to the most suitable BPMN gateway.
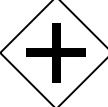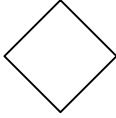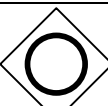
BPMN	SBPMN
Gateway AND	
Gateway Complex	Gateway
Gateway OR	
Gateway Event-based	

Table 2 BPMN Gateways and SBPMN Gateway

The help to the user is emphasized in the elements utilization that allow to the user to recognize in the notation those needs without to require big efforts, for it an example is the icons utilization that indicate the functionality of a familiar way in the elements so called events.

BPMN	SBPMN
Event Cancel	Event Cancel
Event Exception	Event Exception
Event End	Event End
Event Timer	Event Timer
Event Start	Event Start

Table 3 BPMN Events and SBPMN Events

On the other hand we discard the model proposed by the work [22] since this one allowed to the user to define the requirements and functionalities of the application with a textual format. Therefore this information never could be

represented in a formal language.

1)   *Evaluating the notation*

All characteristics of this notation have been evaluated though of two tests. These tests are one about BPMN and another about SBPMN. These tried to evaluate the same aspects with each notation to demonstrate that SBPMN is simpler, easy to learn and use and agile than BPMN. The tests were structured in three blocks: Notation elements identification, Notation elements matching from the needs and Identification of process modeling mean from notation. The tests were realized by 75 people with different levels in the business process modelling. Below we show a figure that represents the total percentage of skills and failures for every notation without considering the division proposed depending on the level of the users.
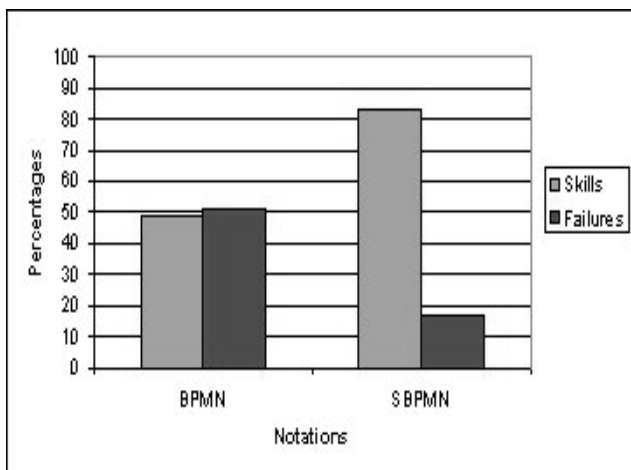

Fig. 5 Graphic of the Global results

Analyzing the results obtained in the three blocks we can establish that the facility of use and understandable of the SBPMN notation is better than in BPMN in all user levels. Even more the users with low and medium levels present more difficulties to understand the use that each one of the elements that BPMN has.

The notation SBPMN represents the most important element of our architecture so that it allows the user to modify his business processes.

B. Necessary Semantics for business process modeling

One of the problems that we initially consider was if we could represent through the BPMN all the necessary semantics. The answer turned out to be affirmative, since to define the necessary semantics of our application we use SBPMN. This notation is a extension of BPMN therefore it has rich and clear semantics since it allows the graphical representation of abnormal execution flows, capture of exceptions, events and compensations as well as the representation of conditions and complex structures, therefore we will be able to reflect the whole semantics of our application as it is commented in [22].
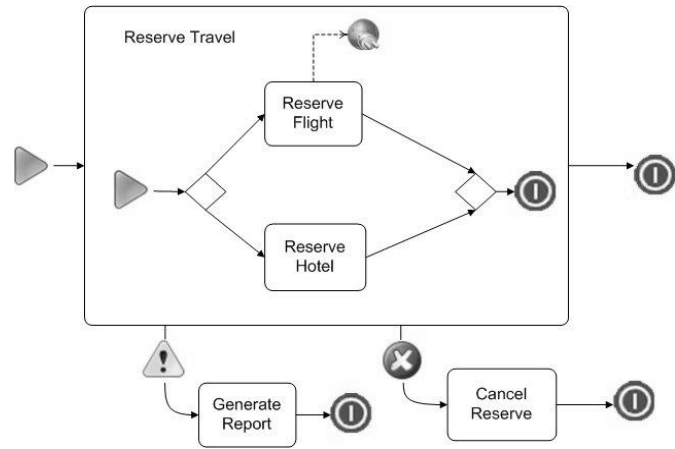

Fig. 5 Example of the notation SBPMN

As it is possible to observe in the figure 4 we can define through SBPMN all the necessary semantics. In this case we can represent the capture of exceptions, flows with condition, utilization of artifacts.

C. Navigation model

One of the initial questions was the navigation model. With the study of the works [10,27] we discovered a methodology to elaborate the navigation model starting from the business process model. In the works [10,27] was commented that in the business processes described in any diagram there are three kind of processes: executed by the user, automatic and external services. All these types of processes are defined in the business process diagram through BPMN, therefore those processes which need human interaction may appear defined in the navigation model.

To carry out the model – model transformation it was established the transformation between the business process model and the navigation model and later it was necessary a second transformation, in this case a model - text transformation which changes the existing navigation model to the presentation technology chosen for the application.

D. Information rendered

In this point we have to bear in mind that as we have an application for an company, they have templates with a determined format for the information rendered, that is why it is drastically reduced the complexity that this point showed at the beginning.

On the other hand for the forms design it will be used XForms [28] that will allow the creation of forms for those processes that need the active partition of the user.

E. Transformations model-model and model-text

In the CIM level it takes place two transformations modelBPMN - modelXPDL and modelXPDL - textPIM. These transformations could have been solved rapidly through the use of BPDM (Business Process Definition Metamodel) [4, 29]. BPDM is a metamodel proposed by

OMG for the MDA paradigm. This metamodel has not been declared as a standard yet but a lot of works so assure that it will finally does. It tries to establish the logical relations between the different types of business process modelled independently of notation or methodology, trying to define the connections between terms and concepts. Therefore any notation like BPMN or the UML 2.0 activity diagrams would be able to use the BPDM metamodel as a bridge to any other metamodel for example to UML class diagrams or to XPDL metamodel.

This technology has not been used due to the fact that there is no official specification and it still has not been adopted as standard by OMG. Therefore to make these transformations ATL has been used defining the transformation rules manually. ATL is a language of transformation model based on the standards OMG [2], MOF [17], QVT [17] and OCL 2.0 (Object Constraint Language). It is a hybrid language since it works with declarative and imperative constructions. The declarative constructions are the option preferred to write transformations, since they are clear and precise. They allow to express correspondences, between the elements of the source model and of the target model, from a series of compositions of rules. Additionally the imperative constructions provide builders to make easy the specification of correspondences that in a declarative way would be much more complex.

1) Transformation from model BPMN to model XPDL

Beginning with the use of ATL as a transformation metamodel between both models we have to emphasize that MDA forces the use of upper models such as metametamodels. In our case we will use MOF, Ecore (Eclipse Modeling Framework). To facilitate the work and avoid the transformation from a KM3 model to a Ecore metametamodel, as it was commented in [31], we decide to use Ecore's existing metametamodels for BPMN 1.0 and XPDL 2.0 that can be downloaded from Eclipse. Finally we use the ATL transformation rules defined between both models obtaining the XPDL model.

```
rule SequenceFlow2Transition{

    from
        b : BPMN!SequenceFlow
    to
        out : XPDL!Transition(

        quantity <- b.quantity,
        from_element<- b.from_element,
        to_element <- b.to_element
        )

}
```

Fig. 6 Example of the file of transformation atl

The objective of using XPDL 2.0 owes especially to its interoperability power and to the critical fact of expressing all the necessary semantics that BPMN owns. For the

creation of this transformation we have born in mind the experiences of the works [30,31], obtaining a complex but efficient transformation.

2) Transformation from model XPDL to model PIM

For this last transformation we have also used ATL and the XPDL model obtained in the previous transformation. Therefore it is only necessary to define the ATL equivalence rules between the XPDL model and the input XML format to the PIM of our MDA. The result is a XML text file that will be the necessary for our PIM level.

```
<PIM>
  <actions>
    <action Id="ACTION_RECEPTION_LECHE" NumberId="1"
    Relation="1_1" Simple="ITEM_RAW_LECHE|ITEM_RECEPTION_LECHE" Multiple="" Must="" Main="" Info="">
      <inputs>
        <input Id="ITEM_RAW_LECHE" />
      </inputs>
      <outputs>
        <output Id="ITEM_RECEPTION_LECHE" />
      </outputs>
      <constraints>
        <constraint ErrorMessage="CONSTRAINT_ACTION_COUNT_=1-1"
        Type="count" Mode="strong">($ITEM_RAW_LECHE $EQUAL 1 $AND $ITEM_RECEPTION_LECHE $EQUAL 1)</constraint>
        <constraint ErrorMessage="CONSTRAINT_ACTION_EXPRESSION_LOCATION_TEMPERATURA"
        Type="expression"
        Mode="strong">($ITEM_RAW_LECHE.PROPERTY_TEMPERATURA $GREATER -100) $AND ($ITEM_RAW_LECHE.PROPERTY_TEMPE
      </constraints>
      <devices>
        <device Id="DEVICE_ETIQUETADORA_1" Interface="WEB" Operation="LABELPRINTER" />
        <device Id="DEVICE_ETIQUETADORA_1" Interface="DEVICE_BIZERBA_1" Operation="LABELPRINTER" />
      </devices>
    </action>
```

Fig. 7 Example of XML of the PIM

IV. GENERATION OF THE APPLICATION

Once overcome the points that represented a difficulty in the creation of the architecture, we can establish that from the transformations commented in the point 3.5 we obtain the complete CIM. The output provided by the CIM corresponds with the XML necessary for the PIM, from the PIM our MDA it is capable to generate all the application. The generation of the application through MDA is commented with more detail in the work [1].

Fig. 8 Example of a screen of the application

V. CONCLUSIÓN

The designed architecture in this paper pretends to offer an system over the current technologies, to be able to allow to the user to modify the behavior of his application through a business process notation simple and understandable without need of technical knowledge. To confront such a challenge we tried to find technologies that were bringing us the most possible over the fulfillment of our goals for this reason we mix the power of BPM and the capacity of applications generation of MDA. These two technologies were bringing us over to the resolution of the problem, but themselves they were not solving it.

Therefore we had that to extend BPMN, to obtain a notation that was understood by the business user, for this reason we created SBPMN.

Later we checked that the pass between the SBPMN diagram and the PIM of our MDA. It was difficult because it required complex transformations and the same time SBPMN would have to be able to represent more aspects as the navigation, semantics.

We think that this architecture represents a point of departure for the business application generation that allows the immediate and simple adjustment the changes without need of costly tasks.

VI. FUTURE WORK

Inside the future work of this paper we need to finish the SBPMN editor of the CIM layer. Also we want to improve those points where we have detected a higher percentages of failures that provoked confusion or the lost of time in the users when they made the tests.

Others of the possible points of investigation it is the information rendered since at the moment we focus on only in the use of existing templates, with what we reduce drastically the possible errors that could arise at the moment of the information rendered beginning with the navigation model generated from the business process model.

REFERENCES

[1] V. García-Díaz, H. Fernández-Fernández, E. Palacios-Gonzalez, C. Pelayo G-Bustelo, J.M. Cueva Lovelle. "Intelligent tracability system of Cabrales cheese using MDA TALISMAN". University of Oviedo, 2007.
[2] Object Management Group, http://www.omg.org/.
[3] Douglas C. Schmidt. "Model-Driven Engineering". Vanderbilt University, Feb-2006.
[4] Harmon Paul. "The OMG´s Model Driven Architecture and BPM". Business Process Trends, Page numbers (1-12), May-2004.
[5] J. Paulo Barros, Luis Gomes. "From Activity Diagrams to Class Diagrams". University Nova of Lisboa, Page numbers (2), 2003.
[6] David Skogan, Roy Gronmo, Ida Solheim. "Web Service Composition in UML". SINTEF Information and Communication Technology, Page numbers (1-12), 2004.
[7] Business Process Modeling. OMG, http://www.bpmn.org/.
[8] Lauri Eloranta, Eero Kallio, Ilkka Terho. "A Notation Evaluation of BPMN and UML Activity Diagrams"; Page Numbers (1-45), 2006.
[9] Juan Diego P.J., Amador D.T., Antonio R.C. "¿Por qué OMG ha elegido BPMN para modelar los Procesos de Negocio si ya existe UML?". University of Salamanca,2007.
[10] Victoria Torres, Vicente Pelechano. "Building Business Process Driven Web Applications". Technical University of Valencia, 2007.
[11] Miguel Ángel Sánchez Vidales, Ana Hermoso García, Luís Joyanes Aguilar. "Una recomendación basada en MDA, BPM y SOA para el desarrollo de software a partir de procesos de negocio en un contexto de Negocio Bajo Demanda". University Pontificia of Salamanca, 2006.
[12] Stephen A. White. "Using BPMN to Model BPEL Process". IBM Corp., 2005.
[13] Join White. "WS-BPEL Extension for People – BPEL4People". IBM and SAP, Jul-2005.
[14] Yi Gao. "BPMN-BPEL Transformation and Roud Trip Engineering". eClarus Software, Page numbers (1-6), 2006.
[15] Alfonso Rodriguez, Eduardo Fernández Medina, Mario Piattini. "Hacia la obtención de Clases de Análisis y Casos de Uso desde modelos de Procesos de Negocio". University of Castilla-La Mancha, 2006.
[16] Wararat Rungworawut, Twittie Senivongse. "Using Ontology Search in the Design of Class Diagram from Business Process Model", 2006.
[17] Meta Object Facility (MOF) Query/View/Transformation Specification, Versión 2.0, OMG, http://www.omg.org/cgi-bin/doc?ptc/2007-07-07.
[18] Henk Jonkers, Maarten W.A. Steen, Lex Heerink, Diederik van Leeuwen. "Bridging BPM and MDE: On the integration of BiZZdesigner and OptimalJ", 2007.
[19] BiZZdesigner. http://www.bizzdesign.nl/html/bizzdesigner.html.
[20] OptimalJ, Compuware. http://www.compuware.com/
[21] Eclipse Modeling Framework. Eclipse EMF, http://www.eclipse.org/modeling/emf/
[22] Michael Hevay. "Keeping BPM Simple for Business Users", Business Process Management Trends, Jan-2006.
[23] Atlas Transformation Language (ATL). http://www.science.univ-nantes.fr/lina/atl/
[24] Workflow Management Coalition, "Process Definition Interface - XML Process Definition Language", version 2.0, http://www.wfmc.org/, Oct-2005.
[25] Wil M.P. van der Aalst. "Patterns and XPDL: A Critical Evaluation of the XML Process Definition Language". Eindhoven University of Technology, 2004.
[26] Robert M. Shapiro. "XPDL 2.0 Integrating Process Interchange & BPMN". WfMC, Apr-2007.
[27] Victoria Torres, Pau Giner, Vicente Pelechano. "Web Application Development Focused on BP Specifications". Technical University of Valencia, 2006.
[28] Richard Cardone, Danny Soroker, Alpana Tiwari. "Using XForms for Simplify Web Programming". IBM, 2004.
[29] OMG Group. "Business Process Management with OMG Specifications". OMG, Oct-2007.
[30] Beatriz Mora, Francisco Ruiz, Félix García, Mario Piattini. "Experiencia en transformación de modelos de procesos de negocios desde BPMN a XPDL". University of Castilla-La Mancha, 2006.
[31] CAPE VISIONS. "XPDL with BPMN Extensión.", 2004.
biography.