# AGILE Web Development using WebBPMN

Riccardo Cognini
Research Unit
e-Linking Online System
Camerino (MC), Italy
Email: riccardo.cognini@e-lios.eu

Alberto Polzonetti
Research Unit
e-Linking Online System
Camerino (MC), Italy
Email: alberto.polzonetti@e-lios.eu

*Abstract*—**In recent years, many web applications regulated by workflows were developerd in order to permit the collaboration between many stakeholders during the execution of Business Processes. Generally, these kind of systems are implemented using web technologies in order to be easily designed, implemented and to be available in different operative systems and platform. Standard general purpose Business Process Modeling languages such as BPMN (Business Process Modeling and Notation) can be used to design the workflow of these systems, but they lack in the definition of which web technology has to be used to implement each single business activity. Another problem due to the complexity of this application is to choose the right software design process to be flexible enought to deal with changes in requirements and in software code itself. In this paper, we propose a novel business process modeling notation named webBPMN including elements that can be used to model web applications regulated by workflows. The notation can be combined with AGILE design process in order to develop flexible web application. We experimented the approach designing the Business Process of the internship web application of the University of Camerino.**

***Keywords–Web Application; Business Process; Metamodel.***

## I. INTRODUCTION

Inter-organizational Business Processes (BPs) permit to different stakeholders to successfully cooperate in order to reach a common target goal [1]. Each stakeholder involved in the BP pursues its objectives within the cooperation and shares its competencies and processes to provide the integrated functionality. Implementing a software system that allows that cooperation among different stakeholders is not an easy task. It requires a deep analysis of requirements, activities flow and how the actors interact.

Furthermore, many software systems are web based applications in which there are many stakeholders that collaborate in the same environment [2]. In several cases, these software systems are based on a BP, it means that stakeholders have to perform in a specific way a predefined set of activities in order to reach goals. For instance, in an e-commerce web site there is a specific BP that drive the buyer and the seller in order to complete the purchasing of goods.

One of the main issues during the design phase of a software system is the definition of the flow of activities that have to be performed by stakeholders [3]. Languages such as Business Process Modeling and Notation (BPMN) [4], UML Activity Diagram (UML AD) [5], Yet Another Workflow Language (YAWL) [6] or EPC [7] are used to define the flow of activities in imperative way. The main problem of these languages is that they are general purpose and do not provide specific elements to design workflows for web based applications.

Another issues in software engineering is to find a suitable way to reach all the requirements and the request of customers. Traditional software engineering design processes such as waterfall or iterative approaches lack in flexibility since they are to structured and they do not react to requirement chenges.

In this paper, we propose webBPMN, a BPMN 2.0 variation, in order to include elements that can be used to design web application based on BP using an AGILE development technique. In particular, we consider new types of tasks and sub-processes assuming that a single atomic activity can be performed in single *Web Page Task* and that a web page can be used in order to perform many activities, then it is a *Web Page Sub-Process*. Other elements to specify client/server side functions and events are also designed. We also comabine webBPMN with the AGILE process in order to design a software that implement a structured procedure but at the same time it is quick and reactive to changes.

The proposed approach was used to design the BP of the internship procedure of the University of Camerino. Its software system was implemented starting from a webBPMN model.

The paper is structured as follow. Section II provides background material related to BP modeling and AGILE. In Section III the webBPMN notation is explained, the in Section IV the approach is described. Then Section V is about related works and, finally, in Section VI we treat some conclusions.

## II. BACKGROUND

In this section background materials about Business Process Management and AGILE are provided.

### A. Business Process Management

Business Process Management (BPM) “includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of Business Processes” [8]. “A BP is a collection of related and structured activities undertaken by one or more organizations in order to pursue some particular goal. Within an organization a BP results in the provisioning of services or in the production of goods for internal or external stakeholders” [9]. Public services structure, their input and output, the interdependencies among different elements can be modeled and implemented using notations and tools supporting the BP abstraction.

The accuracy of the BP modeling phase is critical for the success of an organization in particular in scenarios in

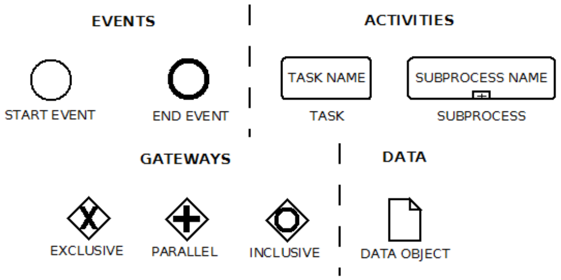

Figure 1. BPMN 2.0 Core Elements.

which it is necessary to adapt to changing requirements. In order to design a BP different classes of languages have been investigated and defined.

In our work, we refer to BPMN 2.0 an Object Management Group standard [4]. It is the most used language by domain experts due to its intuitive graphical notation. We have mainly used process diagrams, focusing on the point of view of system users. The following BPMN 2.0 elements (Figure 1) are the core elements of the language and those we will use on the approach.

- **Events**, which are used to represent something that can happen. An Event can be *Start Event* represents the points in which the BP starts, and *End Event* is raised when the BP terminates. Events are drawn as circles.
- **Activities**, which are used to represent a generic work that a company performs within a BP. An Activity can be atomic - *Task* - or not - *Sub-Process*. Activities are drawn as rectangles with rounded corners.
- **Gateways**, which are used to manage the flow of BP both for parallel activities and choices. Different types of gateways are available, the most used are followed reported. *Parallel Gateway* has to wait all their input flows to start and then all the output paths are started in parallel, it can behave as a fork respects to output paths or as a merge respects to input paths. *Exclusive Gateway* gives the possibility to describe choices both in input and output, it can be activated each time the gateway is reached and when executed it can activate exactly one output path. *Inclusive Gateway* gives the possibility to select among multiple output paths each time they are reached, it can behave also as inclusive merge. Gateways are drawn as diamonds.
- **Data Objects**, which permit to model documents, data, and other objects used and updated during the BP. Objects can also be characterized by a state. An activity can require or can generate a data object in a particular state, whereas if the state is not explicitly reported the activity is state independent. A data object cannot be in two different states at the same time. If the same object is linked to the same activity specifying two different states, this means that states are exclusive with respect to each other, therefore when the activity is executed it needs the data object in one of the available states. A Data Object is represented by a portrait-oriented rectangle that has its upper-right corner folded over. States are represented using text within squared brackets located under the object name.

Using BPMN it is also possible to define the participants (or stakeholders) involved in a BP. *Pools* can be used as elements containers in order to specify the activities that have to be executed by a single participant, they are represented via rectangle containers. Participants can communicate each other using *Message Flow* that specify that a particular task or event can be performed only if a message from another participant is received. *Message Flow* are graphically represented via a dotted arrows. An example of BPMN model in which participants communicates is shown in Figure 2. The BP shows that there are two participants, they are *Participant 1* and *Participant 2*. *Participant 1* starts the execution of the BP and then he sends a message to *Participant 2* executing the task *Send a Message*. *Participant 2* starts the execution of his BP when he receives the message from *Participant 1*, it is why there is a *Start Message Event*. Then, *Participant 2* executes the tasks *Check the Message* and *Responde* in order to send a message to *Participant 1* and ending his BP. When *Participant 1* receives this message, he has to choose if execute the task *Do Action 1* or *Do Action 2*. After the execution of the chosen task the BP execution ends.

## *B. Agile Web Development*

An agile approach to web development is an attitude that promotes adaptation, cross-functionality, and continual collaboration amongst a team. To be agile, programmers and Project Manager must constantly be thinking months in advance and must adapt to any changes that may happen. They are planning early on, meeting with your team in scrum huddles, establishing deliverables, meeting goals ahead of schedule and making continual improvements. It is necessary that the team is completely flexible to changes during the development process. Within this flexibility, the team anticipates changes and respond accordingly, then the team must predict, execute and adapt.

A cornerstone of agile web development is also the scrum process in which the team has a quick meeting, discusses progress and implementation, and then goes their separate ways, while still functioning as an autonomous unit. Afterward, all relevant information is continually relayed to the client and the projects are divided into sprints and user stories. It helps foster proper communication and maintain an ideal agency/client relationship. Adopting an agile process helps eliminate unnecessary wastes of time and allows software companies to allocate those precious minutes and hours on actions and processes that add value to your website. Agile web development can easily be summed up with one word: efficiency. Compared to traditional software engineering, agile software development mainly targets complex systems and product development with dynamic, non-deterministic and non-linear characteristics, where accurate estimates, stable plans, and predictions are often hard to get in early stagesand big up-front designs and arrangements would probably cause a lot of waste, i.e., are not economically sound. These basic arguments and previous industry experiences, learned from years of successes and failures, iterative and evolutionary development [10].

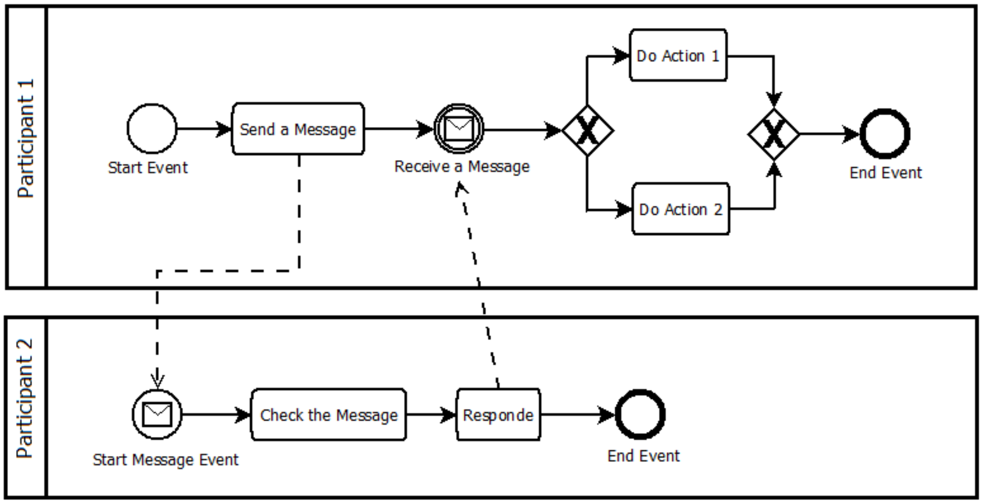

Figure 2. BPMN 2.0 Model Example.

### C. The SCRUM Approach

Scrum appraoch is written from the perspective of the customer. By thinking of what the customer desires/needs, it allows the development team to better understand how to achieve goals for the website. A simple way to look at this is to think of this by using the following sentence: As a (role) I want (feature) so that (benefits). As you collect user stories, you begin to acquire a product backlog, which is a collection of user stories. These helps facilitate direction and milestones. There are three core roles in the Scrum framework. These core roles are ideally collocated to deliver potentially shippable Product Increments. They represent the Scrum Team. Although other roles involved with product development may be encountered, Scrum does not define any team roles other than those described below.

- **Product Ownner** The Product Owner represents the product's stakeholders and the voice of the customer; and is accountable for ensuring that the team delivers value to the business. The Product Owner writes customer-centric items (typically user stories), prioritizes them based on importance and dependencies, and adds them to the Product Backlog. Scrum Teams should have one Product Owner [11].
- **Development Team** The Development Team is responsible for delivering potentially shippable increments (PSIs) of product at the end of each Sprint (the Sprint goal). A team is made up of 39 individuals who do the actual work (analyse, design, develop, test, technical communication, document, etc.). Development Teams are cross-functional, with all the skills as a team necessary to create a Product Increment. The Development Team in Scrum is self-organizing, even though there may be some interaction with a project management office.
- **Scrum Master** Scrum is facilitated by a Scrum Master, who is accountable for removing impediments to the ability of the team to deliver the product goals and deliverables. The Scrum Master is not a traditional team lead or project manager, but acts as a buffer between the team and any distracting influences. The Scrum Master ensures that the Scrum framework is followed. The Scrum Master helps to ensure the team follows the agreed processes in the Scrum framework, often facilitates key sessions, and encourages the team to improve. The role has also been referred to as a team facilitator or servant-leader to reinforce these dual perspectives [12]

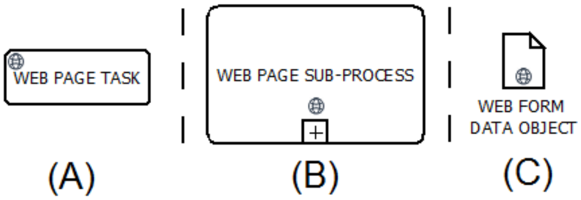

Figure 3. webBPMN Main Elements.

## III. WebBPMN

The proposed language named webBPMN is a BPMN 2.0 variation in which standard BPMN tasks and sub-processes are replaced with activities designed for web based applications implementing workflows. A new type of data object is also introduced in order to model the type of data used by web pages to communicate. These kinds of elements are described as follow.

- **Web Page Task** is an atomic activity performed in a generic single web page (Figure 3-A). When the activity is performed a new page will be open. In this kind of task the stakeholder has to interact with the software system via a web page. For instance, a *Web Page Task* can be related to a search page or a form that the stakeholder has to fill;
- **Web Page Sub-Process** is a composed activity performed in a complex web page (Figure 3-B). In this kind of activity the stakeholder should perform more that one activities in a single page beside client side technologies (such as Javascript/AJAX) or/and some

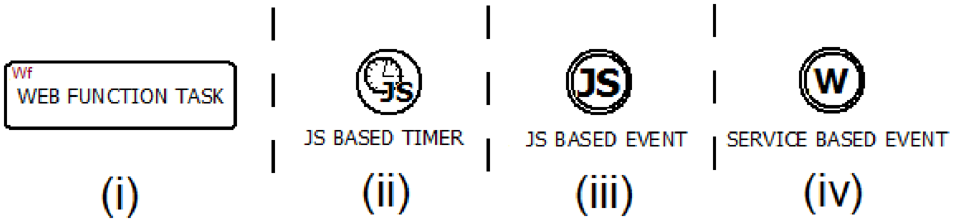

Figure 4. webBPMN Elements that can be used in a Web Page Sub-Process.

events can happen. For instance a *Web Page Sub-Process* can be used to represent a page in which a web chat is implemented. A *Web Page Sub-Process* may contain a set of specific BP elements that are able to specify the behaviours of the activity. Such as in many others BPMN sub-processes a start and and end event should be insert. The list of elements that can be inserted inside are described as follow.

- ○ **Web Function Task** is an atomic activity that can be performed in a web page without open a new page (Figure 4-i). It can be performed automatically or manually by an user, it can be client or server side. For instance, it can be used to call a web service in asynchronous way or to perform any server side request via AJAX;
- ○ **JavaScript Based Timer** is an event that is trigged by a timer for 1 or more times (Figure 4-ii). For instance, it can be used to refresh the page or to update the list of messages and users connected in a web chat;
- ○ **JavaScript Based Event** is a generic client side event that can be trigged by an user action such as change the value in a dropdown list (Figure 4-iii). For instance, it can be used to represent the event that is trigged when there is an error in a Web Form input field. When an event like that is trigged a *Web Function Task* should be performed;
- ○ **Service Based Event** is a generic event that is trigged when a Web Service respondes to a call (Figure 4-iv). For instance, it can be used to start activities when a login server responds to a request;

- **Web Form Data Object** is a particular type of data that can be generated by a web form (Figure 3-C), it means that it can be generated only by a *Web Page Task* or a *Web Page Sub-Process*. A *Web Form Data Object* is composed by a set of values and eventually by a set of data files. This data object is generated by a web page and it can be consumed by another web page. For instance, this data object can be generated by a form in which an user has to log into the web application and then it can be consumed in another page in order to verify the credentials.

Using webBPMN a BP designer can use all the standard BPMN 2.0 elements except tasks and sub-processes. Pools should be used to define which web pages can be accessible by a specific stakeholders. The interaction between stakeholders can be defined via standard BPMN *Message Flows*. Gateways are used to drive the route of the flow and define which pages should be open for each stakeholder.

## IV. The Approach

The approach that we proposed expects three different steps.

- The first step is performed by Software designer with skills in BP modeling. Using a story telling approach, he collects all the requirement by the customers. The requirement will be grouped in order to divide the work in different modules. Each modul can be implemented by one or more programmers. To be Agile in this step, the Software designer should divide the requirements using the SCRUM approach [13]. In this specific context the Software designer can be considered the SCRUM Master.
- In the second step, the Software Designer designer model the webBPMN model of the web application. From this point until the end of the development he will work as SCRUM Master, so he will remove impediments that afflict the development team.
- In the last step, starting from the webBPMN models the development team starts to implement each singol task. After each task implementation there is a meeting with the Product Ownner in order to evaluate the work that was done.

### A. Use Case

The described approach has been applied to model the Web Application of the Students Internship Business Process of the University of Camerino. This is a service that the Univeristy has to put in place in order to permit studens to start an internship in Italian Companies - in University of Camerino each Bachelor student must do an internship to graduate. The proposed webBPMN is shown in Figure 5, it is just a simplied version of the real one.

Three stakeholders are involved in this business process, the student that has to apply for an internship, the related company and the Internship office of the University.

The trigger of a process instance is the student that has to do an internship. First, the student accesses to the Login Page that is modeled as a Web Page Sub-Process since several function are needed to verify the identity of the student. The Web Function Task *Request ESSE3 credentias* is delegated to request username and password of the student, then the Web Function Task *Connect to ESSE3 (LDAP Server)* connects the user to ESSE3 system in which credentials are stored. The

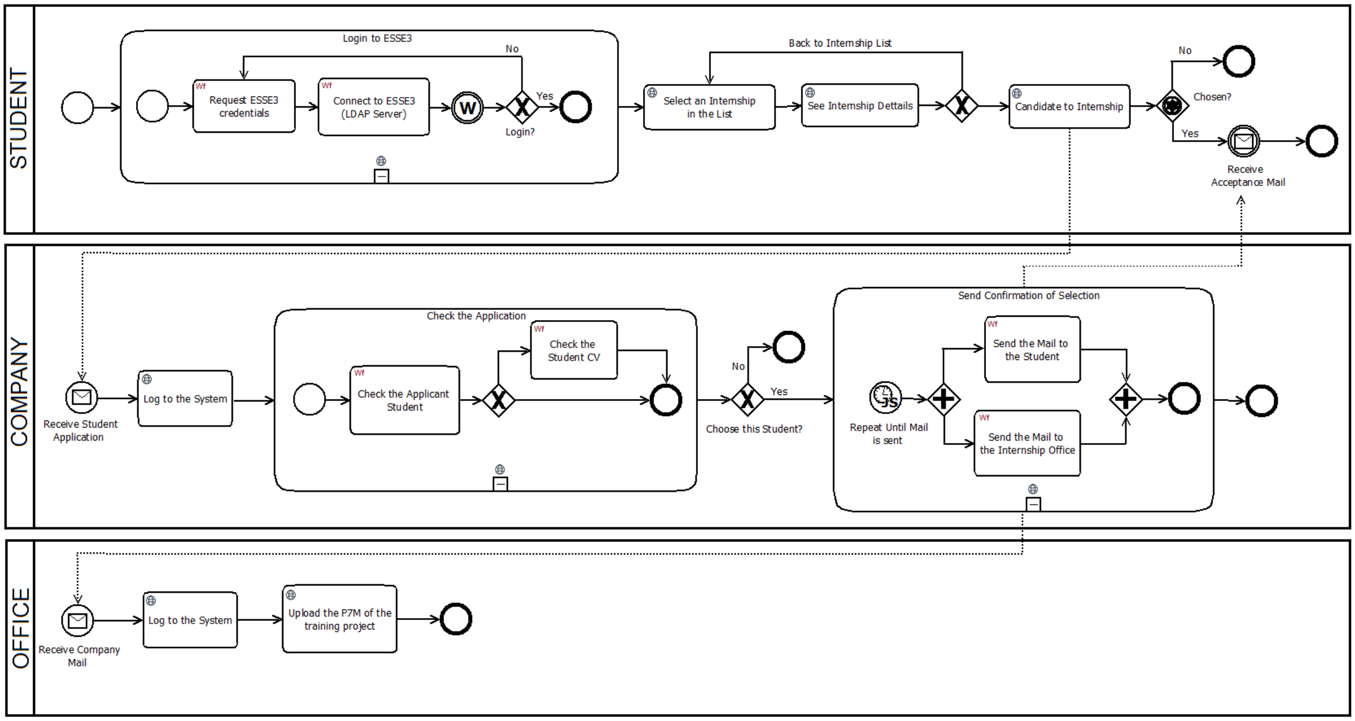
STUDENT
Login to ESSE3
No
Request ESSE3 credentials
Connect to ESSE3 (LDAP Server)
Yes
Login?
Back to Internship List
Select an Internship in the List
See Internship Dettails
Candidate to Internship
No
Chosen?
Yes
Receive Acceptance Mail
COMPANY
Receive Student Application
Log to the System
Check the Application
Check the Applicant Student
Check the Student CV
No
Yes
Choose this Student?
Send Confirmation of Selection
Repeat Until Mail is sent
Send the Mail to the Student
Send the Mail to the Internship Office
OFFICE
Receive Company Mail
Log to the System
Upload the P7M of the training project

Figure 5. Internship BP of University of Camerino.

response arrives in the Service Based Event, if the credentials are correct the page related to the *Select an Internship in the List* Web Page Task is open. In this page, the student can see the list of available internships, he/she can choose one of them in order to open the detail page of the internship specified by the *See Internship Details* Web Task Page. The student can candidate to the internship or return to the page in which there is the list.

When the student applies for an internship the related company receives a communication. Then, the company will log to the system via the *Log to the System* Web Page Task in order to check the student application in the *Check Application* Web Page Sub-Process. In this page the company checks the information about the student (*Check the Applicant Student* Web Function Task) and, optionally his CV (*Check the Student CV* Web Function Task). The company can choose the student for the internship of not, in case the student is chosen the *Send Confirmation of Selection* Web Page Sub-Process is open. In this page, e-mails are sent to the student and to the Internship office of the University via *Send the Mail to the Student* and *Send the Mail to the Internship Office* Web Function Tasks. These tasks are performed automatically by an JavaScript Based Timer that repeat the execution until mails are sent correctly.

When the company chooses a student for an internship, the Internship office receives a mail. The office has to log to the system via the *Log to the System* Web Page Task and then upload the P7M file related to the internship via the *Upload the P7M of the training project* Web Page Task.

Notwithstanding the complexity of the Web Application modeling of the scenario has revealed that webBPMN permits to define which activities have to be performed in each web page.

## V. Related Work

In literature, there are just few languages to model web applications. Usually, languages provide few models to design different requirements of the web the applications.

The model Object Oriented Hypermedia (OO-H) is used to design generic web applications [14]. In particular, OO-H provides a navigation model named Navigation Access Diagram (NAD) that provides the necessary constructs to represent how web site user navigates between links. It is similar to the concept explained in this paper in which web pages are represented as business activities, but the used semantics is not able to represent the activities that can be performed in each web page. There are some language similar to OO-H such as Object-Oriented Hypermedia Design Method (OOHDM) [15], they share the same base approach.

Another interesting language is UML-based Web Engineering (UWE) that is an object oriented and iterative approach based on UML [16]. Also in this language, there is a navigation model to specify how users navigate between links of the web site. Also in this case, the activities that users have to perform in each page are not defined by the model.

Web Modeling Language (WebML) is a language to design web sites [17]. It provides an high-level graphical representation providing different models in order to design several aspects of the designed web application. In this case there is the Composition Model that specifies the pages provided by the web application, instead the Navigation Model specifies how pages are linked together.

The main issue of these language is that they are not focused on the BP modeling. They can be used to design a generic web site also if it is not regulated by a BP. It means that they do not provide a full specification to manage all the possible situations that can happen in a BP. Instead, thanks to BPMN base notation webBPMN provides a set of elements that can be used to describe many situation and, thanks to the variation of the semantics of the BPMN activities it is possible to adapt the language for the web application modeling.

A language similar to webBPMN is WebWorkFlow that is an object oriented workflow modeling language [18]. It can be used to design simple activities flows for web application. The main lack of the language is that it does not provide a graphical representation since it is mainly used to generate an executable application. It means than also the elements provided by the language are not so much and their semantics are more related to programming languages than BPs.

## VI. Conclusion and Further Work

In this paper, we present an approach that combine a BP modeling language with AGILE approach in order to design web application in more efficient wway. The BP notation proposed seems particularly suitable to specify which functions have to be implemented in which web page and the AGILE approach seems to be very useful to understand better the requirements and the needs of the customers. The experiment related to University of Camerino Internship BP provides encouraging results.

In the next future we plan to extend the webBPMN notation including new elements in order to specify in a better way web technologies and functions. For instance, we will include the concepts of Session and Cookies extending again the BPMN data objects. We are implementing a modeling environment to use the webBPMN notation, we are using ADOxx (http://adoxx.org/) to design the webBPMN meta-model. At the end, we will continue the experimental work considering other use cases in the e-government scenario.

## Acknowledgment

The authors would like to thank the University of Camerino.

## References

[1] V. Vathanophas, “Business process approach towards an inter-organizational enterprise system,” Business Process Management Journal, vol. 13, no. 3, 2007, pp. 433–450.

[2] A. Ginige and S. Murugesan, “Web engineering: an introduction,” MultiMedia, IEEE, vol. 8, no. 1, Jan 2001, pp. 14–18.

[3] M. Dumas, W. M. van der Aalst, and A. H. ter Hofstede, Process-aware Information Systems: Bridging People and Software Through Process Technology. New York, NY, USA: John Wiley & Sons, Inc., 2005.

[4] B. P. M. OMG, “Notation (bpmn) version 2.0 (2011),” Available on: http://www. omg. org/spec/BPMN/2.0, 2011.

[5] OMG, OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, Object Management Group Std., Rev. 2.4.1, August 2011. [Online]. Available: http://www.omg.org/spec/UML/2.4.1

[6] W. M. Van Der Aalst and A. H. Ter Hofstede, “Yawl: yet another workflow language,” Information systems, vol. 30, no. 4, 2005, pp. 245–275.

[7] J. Mendling, “Event-driven process chains (epc),” in Metrics for Process Models, ser. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2008, vol. 6, pp. 17–57. [Online].

[8] M. Weske, Business process management concepts, languages, architectures, 1st ed. Springer, Nov. 2007.

[9] A. Lindsay, D. Downs, and K. Lunn, “Business process - attempts to find a definition,” Information and Software Technology, vol. 45, 2003, pp. 1015–1019.

[10] C. Larman, Agile and iterative development: a manager’s guide. Addison-Wesley Professional, 2004.

[11] R. Pichler, Agile Product Management with Scrum: Creating Products that Customers Love (Adobe Reader). Addison-Wesley Professional, 2010.

[12] E. Leybourn, Directing the Agile organisation: A lean approach to business management. IT Governance Ltd, 2013.

[13] J. Sutherland and K. Schwaber, “The scrum guide,” The Definitive Guide to Scrum: The Rules of the Game. Scrum. org, 2013.

[14] P. Plessers, O. D. Troyer, and S. Casteleyn, “Event-based modeling of evolution for semantic-driven systems,” in Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal, June 13-17, 2005, Proceedings, 2005, pp. 63–76. [Online]. Available:

[15] D. Schwabe and G. Rossi, “An object oriented approach to web-based applications design,” Theor. Pract. Object Syst., vol. 4, no. 4, Oct. 1998, pp. 207–225. [Online]. Available: http://dx.doi.org/10.1002/(SICI)1096-9942(1998)4:4¡207::AID-TAPO2¿3.0.CO;2-2

[16] N. Koch and A. Kraus, “Towards a common metamodel for the development of web applications,” in Web Engineering, ser. Lecture Notes in Computer Science, J. Lovelle, B. Rodrguez, J. Gayo, M. del Puerto Paule Ruiz, and L. Aguilar, Eds. Springer Berlin Heidelberg, 2003, vol. 2722, pp. 497–506. [Online]. Available:

[17] S. Ceri, P. Fraternali, and A. Bongio, “Web modeling language (webml): a modeling language for designing web sites,” Computer Networks, vol. 33, no. 1, 2000, pp. 137–157.

[18] Z. Hemel, R. Verhaaf, and E. Visser, “Webworkflow: an object-oriented workflow modeling language for web applications,” in Model Driven Engineering Languages and Systems. Springer, 2008, pp. 113–127.