An empirical evaluation for business process tools

Erika M. Nieto-Ariza[1], Guillermo Rodríguez-Ortiz[1,2], Javier Ortiz-Hernández[1]
1 Centro Nacional de Investigación y Desarrollo Tecnológico,
Interior internado Palmira s/n, Cuernavaca, Morelos, 62490 México
{erika, ortiz}@ cenidet.edu.mx
Home page: http://www.cenidet.edu.mx
2 Instituto de Investigaciones Eléctricas, Reforma 113, 62490,
Cuernavaca, Morelos, 62490, México
gro@iie.org.mx

Abstract. As the use of web grows, organizations are increasingly choosing to use it to provide their services. The modeling process is a previous step in the systematization of a process. Due to the great number of modeling tools in existence, it is necessary to identify the information that tools allow to specify. A set of concepts is proposed to evaluate modeling tools using three levels of abstractions. The proposal compares the modeling capabilities supplied by the different techniques and allows determining what modeling tool is the most appropriate to model specific concepts of interest to a problem.

1 Introduction

Models are commonly used to represent complex systems and to observe the performance in the business process when a technology system is integrated. Technology systems should support business and they become an integral part of the business process [1,2,3,4,5]. Due to the great number of techniques to model and specify requirements, it is complex and laborious to compare them. Three modeling levels are proposed which integrate a set of concepts to build web application models: a) Organizational, its goal is to describe how the organization works and the business process that are going to be systematized with a web information system; b) Conceptual, its goal is to describe the role of the software system and its integration with a particular organizational environment; c) Web, its goal is to describe the semantics of a web application [5,6]. The basis of our contribution is in the detection and classification of a set of concepts which are used to analyze, to evaluate modeling tools and to recognize the capabilities that each tool has in order to model at the three levels of abstraction.

There are some methods and methodologies to evaluate business process modeling, but they evaluate the functionality of an application or a modeling tool. Rosemman proposes an ontology to evaluate organizational modeling grammars identifying their strength and weaknesses [7]. Luis Olsina and Devanshu Dhyani [8, 9] propose a methodology to evaluate the characteristics of a web application in operational phases. The structure of this paper is as follows: in section 2 the modeling concepts that comprise our approach are presented, in section 3 the modeling concepts are enhanced with a set of aspects found to be useful in building models, in section 4 the evaluation results are presented, in section 5 a product evaluation is presented, last the conclusions are discussed.

2 Modeling concepts

A business process model can be viewed at many levels of abstraction, and complementary model views can be combined to give a more intelligible, accurate view of a system to develop than a single model alone [3]. This approach establishes three levels of abstraction and each one includes certain modeling concepts of features (table 1). Concepts are properties or characteristics that structurally describe types of requirements and define the key elements in a business process. The concepts facilitate integration of the levels of abstraction, such that, starting with an organizational model, the elements of the conceptual and the web model are easily identified. The selection of the concepts is a task that requires the analysis of different modeling tools. Through the correspondence of an concept in one level to its corresponding concept in the next level, the three levels are integrated in a complete view of the business process. For example, the task concept in the organizational level correspond to the functional concept at the conceptual level and later it will be correspond to an event concept at the Web level of abstraction.

Table 1. Modeling concepts at each level of abstraction

Organizational level	Conceptual level	Web level	
		Business process	Pure navigation
Actor	Actor	---	Navigation page – Relationship
		User profile (Rol)	User profile (Rol)
		Class (objetct)	---
Resource	Artifact	Artifact	Artifact
Goal	Goal	---	Objective
Task	Function	Service	Service
Activity	Event	Event	---
Business rule	Constraint	Pre and post condition	---
Quality	No functional requirement	No functional requirement	---

The **organizational** modeling concepts are as follows.
- *Goal*. It describes a business process desired state that an organization imposes to itself, with a certain degree of priority; the goal must be quantified whenever possible.
- *Actor*. It describes an entity that has a specific goal, participates in the business process, or has relationships with other actors. An actor may have different roles.
todo- *Resource*. It describes an informational or physical entity that is transferred between actors as a result of task executed by an actor.
- *Task*. It describes a series of activities oriented to reach a goal; it may indicate how should be accomplished.
- *Activity*. It describes a set of actions to carry out one task.
- *Quality*. It describes the desired characteristics in the business process.
- *Business rule*. It describes the actions and criteria that govern the execution of the business process.

The **conceptual** modeling concepts are as follows.
- *Goal*. It describes the information system purpose, limitations and responsibilities, from the business view point.
-*Actor*. It describes an entity (human, hardware, software or process activity) that interacts with the information system and that might play different roles.
- *Artifact*. It describes an abstract or physical entity that is transferred between an actor and the information system.
- *Function*. It describes a service that must be provided by the system to the actors.
- *Event*. It describes a change in the business process in one instant specific of time.
- *Non functional*. It describes the desired quality features or constraints for the information system as for example, platform and interface requirements, etc.
- *Constraint*. It describes a condition for a service execution provide by the system.

The **Web** modeling concepts are as follows.
- *Objective*. The purpose of the Web application, from a simple information pages displayer to a complex and sophisticated corporate portal.
- *Navigation relationship*. It describes a global vision of the Web application according to a user profile with relation to the information to be presented.
- *User profile*. It describes the user unique use of the Web application. A user can have many profiles for the same Web application.
- *Class*. It describes an object type to model the entities that integrate the application, and the information handling for the users to navigate.
- *Artifact*. It describes an abstract object to be transferred between the Web application and a user or vice versa as a result of an event execution.
- *Service*. It describes an activity or an action that the web application has.
- *Event*. It describes the trigger of an activity or action that might be carried out to obtain a result or artifact.
- *Non functional*. It describes the quality features or constrains for the web application.
- *Pre and pos condition*. It describes the performance of an event execution where a precondition is a required object state before the event can be executed and a post condition is the required object state after the event execution.

3 The concepts and the evaluation methodology

The concepts are enhanced with aspects that make them more powerful to model a particular view. These concepts are also used as scales to evaluate modeling tools. The definition of an evaluation scale for each concept is a task that requires the analysis of different modeling tools.

Table 2. Concepts and evaluation scales for the organizational level of abstraction

Scale / Concept	1	2	3	4	5
Actor	Actor	---	Role	Type	Responsibility
Resource	Resource	Type	Actor using it	---	Actor supplying it
Goal	Goal	Priority	Problem	Opportunity	Verification
Task	Task	Who requests	Who executes	Hierarchy	Associated Goal.
Activity	Activity	Tasks supported	Hierarchy	How is activated	When is concluded
Business rule	Business rule	Associated concept	Origin	Type	Hierarchy
Quality	Quality	Associated concept	---	Origin	Measure

Table 3. Aspects and evaluation scales for the conceptual level of abstraction

Scale / Concept	1	2	3	4	5
Actor	Actor	---	Role	Type	Responsibility
Artifact	Artifact	Actor or function supplying	---	Actor or function requiring	Artifact state
Goal	Goal	Who establish it, Associated to a function	Assigned priority	Measure, Failure cause	Opportunity to solve a problem
Function	Function	Who starts it	Who uses it	Hierarchy	The product
Event	Event	Who fires it, What is the start state,	What is produced, Hierarchy	Who receives the product, Owner function	Final state
Constraint	Constraint	Type	Who defines it	To who or what applies	Who or what enforces it
Non functional requirement	Constraint	Who proposes it To what is applied.	Type of requirement.	Measure to verify compliance.	What happens if not fulfilled.

Table 4. Concepts and evaluation scales for the web level of abstraction

Concept \ Scale	1	2	3	4	5
Navigation page - Relationship	Navigation page	Nav. page - Relationship	User Profile	Navigation help	Access constraints
User profile (Role)	User profile	Role	Role changes allowed	Services per user	Business process state
Class (object)	Class (objct)	Attributes	Relationships	Methods	Tye of relationships
Artifact	Artifact	---	Type	Supplier	User
Goal	Who defines it	Associated service,	Priority	Measure	Failure cause, Opportunity to solve it
Service	Related events	Hierarchy, Requesting User	Executing agent, Result.	Result final user	Owner page
Event	Event	Service owner, Hierarchy,	Implementing class	Who requests	Shared or not
Pre and post condition	Post condition	Pre condition	---	---	Associated event
Non functional requirement	Non functional requirement	Who proposes it, To what is applied.	Type of requirement.	Measure to verify compliance	What happens if not fulfilled.

The scale is defined for each concept using the capabilities related to the concept. Also, a desired capability mentioned in the literature may be used in the definition of a scale. Following a well-known approach from the economics and management disciplines [10], to each concept a scale between 0 and 5 is assigned which is going to be used to evaluate one of the modeling capabilities. The order assigned to the scales is intuitive and relatively arbitrary; however, it can be changed easily. The concepts evaluation scales facilitate the comparison of different modeling tools capabilities (see table 2, 3 and 4). The evaluation scale is obtained by first taking a list of the capabilities of one tool, and then a list of capabilities from a second tool, from a third, until all selected tools are analyzed.

The evaluators have to evaluate the three levels of abstraction for all concepts. For each modeling tool and for each aspect a_i, a corresponding evaluation e_i is obtained. The results are displayed in a table for easy of comparison and a total score is obtained for each tool and for each level of abstraction as Σe_i. A tool that scores better than other it possibly has more capabilities to model requirements at the corresponding level of abstraction than the other. The methodology assigns a value to each concept of the method. For example, the precondition and post condition concept at the web level of abstraction; if the method has the post condition aspect, it will have 1 point. If the method has also the precondition aspect, it will have 2 points. If the method has the post condition, precondition and the associated event aspect, it will have 5 points.

4 Evaluation results

To evaluate the scale the following tools were evaluated (tables 5, 6, 7a and 7b): i*, Tropos, EKD, BPM-UML, NDT, OO-Method/OOWS, and OOWS [5, 7, 4, 8, 9, 10, 11, 12, 16]. At organizational level, BPM-UML obtains good scores for this level of abstraction, and i* has the lowest score.

Table 5. Organizational level evaluation of the tools

Organizational level	Max. Value	I*	Tropos	EKD	BPM-UML
Actor	5	5	5	5	5
Resource	5	5	5	2	5
Goal	5	1	3	4	3
Task	5	2	4	3	2
Activity	5	0	2	0	4
Business rule	5	2	0	5	4
Quality	5	3	4	4	4
Total	**35**	18	23	23	27

Table 6. Organizational level evaluation of the tools

Conceptual level	Max. Value	I*	Tropos	NDT	EKD	BPM-UML	OO-Method
Actor	5	5	5	5	5	5	1
Artifact	5	5	5	1	4	5	4
Goal	5	1	3	2	4	3	1
Function	5	2	2	4	5	5	2
Event	5	0	1	2	0	4	3
Constrain	5	2	0	4	5	4	5
No functional	5	3	4	3	4	4	0
Total	**35**	17	20	21	27	30	16

Table 7(a). Web level evaluation of the tools (business process)

Web level	Max. Value	Tropos	OO-Method / OOWS	NDT	OOWS
User profile	5	3	4	3	4
Class	5	0	5	5	5
Artifact	5	4	4	1	4
Service	5	3	3	4	3
Event	5	1	3	2	2
Pre and post condition	5	2	5	4	3
No functional	5	3	0	3	0
Total	**35**	16	24	22	21

Table 8(a). Web level evaluation of the tools (pure navigation)

Web level	Max. Value	Tropos	OO-Method / OOWS	NDT	OOWS
Navegational page – relationship	5	1	5	5	5
User profile	5	3	4	3	4
Goal	5	3	0	2	0
Artifact	5	4	4	1	4
Service	5	3	3	4	3
Total	**25**	14	16	15	16

The tools were evaluated with respect to the parameters defined for the approach presented here. During the evaluation of tools, their own characteristics are shown, for example, the quality aspects of a business process are modeled as qualitative goals using BPM-UML. At conceptual level, the result shows the capacities of each tool, for example, EKD obtains good scores for this level, but OO-Method has the lowest score. At web level, the result shows the capacities of each tool, for example, OO-Method/OOWS obtains good scores for this level, but Tropos has the lowest score.

5 Evaluation methodology of products

Concepts allow to evaluate the products of different tools when they are applied to a specific problem. To show the use, a case study was applied to the i*, Tropos, EKD and BPM-UML tools. The products of these tools were evaluated with the methodology of products. The evaluation capability can be completed with the product evaluation. A brief example of the product methodology is presented. The variables defined for the analysis and evaluation of the products are the following: a) workflow, b) order execution in the function, c) tree of decomposition, d) organization, and e) clear identification of the elements. To each variable a value 0 or 5 is assigned, 5 if the tool has the variable or 0 if it has not the variable. The values assigned to the variables are relatively arbitrary; however, it can be changed. The results in the product evaluation of the tools are presented in the table 8. This evaluation shows that BPM-UML has good score, but in the product evaluation EKD has the best score. The product is an additional reference to select a modeling tool (capability – product).

Table 8. Product evaluation

	Work flow	Order execution	Tree of decomposition	Organization	Identification of elements	Total
I*	5	0	5	0	0	10
Tropos	5	0	5	0	5	15
EKD	5	5	5	5	5	**25**
BPM	5	0	5	5	5	20

Conclusion

There are many proposals to model requirements and each one has its own elements. Some use the same concepts but the names are different, which makes it complex and laborious to compare the tools. The approach presented here unifies the various terminologies, increases the knowledge about modeling concepts, and proposes an evaluation approach for the tools modeling capabilities and techniques. This helps to select the tool that is more appropriate to the needs of a problem domain. Additionally, the approach evaluates the products when different tools are applied to a definition problem. A set of variables is proposed to evaluate the complexity of each model. This helps to know how many capacities the tools has, and also how complex the models are when a specific tool is used. A future work is use metrics on the products or models when different tools are applied. The approach has been used to evaluate e-learning systems [16]. Additionally, it has been applied in the development of various study cases to evaluate virtual reality tools and to clearly appreciate the concepts that the tools allow to model.

References

1. James Pasley, “How BPEKL and SOA are changing web services development”, IEEE Internet Computing. May – June 2005.
2. Peter F. Green, Michael Rosemann y Marta Indulska, “Ontological Evaluation of Enterprisee systems Interoperability Using ebXML”, IEEE Transactions on Knowledge and Data Engineering, Vol 17, No. 5, IEEE Computer Society, may 2005.
3. Mersevy T. and Fenstermacher K., “Transforming software development: and MDA road map”, IEEE Computer Society, September 2005.
4. H. E. Eriksson and M. Penker, Bussiness, Modeling with UML, Chichester, UK, Wiley Editorial, 2000.
5. E. Yu, Modelling Strategic Relation for Process Reengineering, Universidad de Toronto, Canada, 1995. Thesis submitted for the degree of Doctor of Philosophy.
6. A. Ginige and S. M. “Web Engineering: An Introduction” IEEE Multimedia, pp 1-5, Jan-Mar 2001.
7. Peter F. Green, Michael Rosemann y Marta Indulska, “Ontological Evaluation of Enterprisee systems Interoperability Using ebXML”, IEEE Transactions on Knowledge and Data Engineering, Vol 17, No. 5, IEEE Computer Society, may 2005.
8. Olsina, Luis A., Metodología cuantitativa para la evaluación y comparación de la calidad de sitios web. Tesis doctoral. Fac. de Ciencias Exactas, Univ. Nacional de La Plata, noviembre de 1999.
9. Devanshu Dhyani, Wee Keong Ng, and Sourav S. Bhowmick, A survey of web metrics, ACM computer survey, Vol 34, No. 4. December 2002, pp. 469-503.
10. Bubenko J., Brash D. y Stirna J. EKD User Guide, Royal Institute of technology (KTH) and Stockholm University, Stockholm, Sweden, Dept. of Computer and Systems Sciences, 1998.
11. M. J. Escalona, J. torres, M. Mejías, A. M. Reina. From the requirement to the conceptual model in NDT. III Taller de Ingeniería del Software Orientado a la Web Alicante, Spain. November, 2003
12. E. Insfrán, O.Pastor y R. Wieringa, “Requirements Engineering-Based conceptual Modelling”, Requirements Engineering Springer-Verlang, vol. 2, pp. 7:61-72, 2002.
13. J. Gómez, C. Cachero and O. Pastor, “Conceptual modeling of device-independent Web applications” IEEE Multimedia, vol. 8 issue: 2 , pp 26-39, April-June 2001.
14. L. Liu, E. Yu Intentional Modeling to support Identity Management 23rd Int. Conference on Conceptual Modeling (ER 2004). Shanghai, China, November, 2004. Springer. pp. 555-566.
15. J. Fons, O. Pastor, P. Valderas y M. Ruiz, OOWS: Un método de producción de software en ambientes web. 2005. http://oomethod.dsic.upv.es/anonimo/..%5Cfiles%5CBookChapter%5Cfons02b.pdf
16. Eduardo Islas P., Eric Zabre B. y Miguel Pérez R., “Evaluación de herramientas de software y hardware para el desarrollo de aplicaciones de realidad virtual”, consultado en el 2005, http://www.iie.org.mx/boletin022004/tenden2.pdf