INTEGRATING AGILE AND MODEL-DRIVEN PRACTICES IN A METHODOLOGICAL FRAMEWORK FOR THE WEB INFORMATION SYSTEMS DEVELOPMENT

Paloma Cáceres, Esperanza Marcos, Valeria de Castro
Rey Juan Carlos University

Keywords: Agile methodologies, model-driven methodologies, Web engineering.

Abstract: Nowadays, the Web information systems (WIS) development has become an interesting area in both research and business world. On the one hand, the WIS development needs specific methodologies for taking into account certain specific aspects for Web systems. Therefore, traditional methodologies do not facilitate a quick and light development as the agile methodologies. On the other hand, the system modeling becomes too specific because new technologies are emerging and becoming popular constantly. In this way, OMG proposes the Model-driven Architecture (MDA), a model-driven framework for software development. Due to the advantages of both agile and model-driven proposals, a topic of interest is the approach between them. In this way, we are working in a model-driven methodological framework for agile development of WIS, named MIDAS which is presented in this paper.

1 INTRODUCTION

Nowadays, Web Information Systems (WIS) development has become an interesting area in both research and business world.

On the one hand, the WIS development needs specific methodologies (Fraternali, 2000) for taking into account specific aspects of Web systems (Lowe and Hall, 1999). Therefore, traditional methodologies do not facilitate a quick and light development (Fowler, 2000). This is the reason why the agile methodologies (AM) have appeared, which provide a sufficient development process but not excessive.

On the other hand, the system modeling becomes too specific because new technologies are emerging and becoming popular constantly. In this way, the Object Management Group (OMG) proposes the Model-Driven Architecture (MDA) (Miller and Mukerji, 2001), a model-driven framework for software development that proposes to specify the whole system, with platform independent models (PIM) to be able to develop the software system for different platforms. In fact, Model-Driven Development (MDD) has improved productivity, quality and platform independence in the development world, as Kulkarni and Reddy (2003) states.

Nowadays, an interesting topic is the approach between agile and model-driven proposals (Turk et al., 2002), collecting the advantages of both. In this way, we are working in MIDAS (Marcos et al., 2001a, 2001b, 2002, 2003a, 2003b; Cáceres and Marcos, 2001; Cáceres et al., 2003; Vela and Marcos, 2003a; Vela and Marcos, 2003b), a *model-driven* methodological framework for *agile* development of *any kind* of WIS, which takes into account next issues: first, the need of specific methodologies for developing WIS; second, the agile methodologies motivates more than heavy methodologies; finally, modeling first in an independent way and later in a specific technology.

In this paper, we present a preliminar approach for integrating agile and model-driven practices in the MIDAS framework. This approach is based on five aspects which are very differents between both practices.

The rest of the paper is organized as follows: in section 2, the MIDAS framework is presented; in section 3, the integration of the model-driven Web development with agile practices is showed; section 4 sums up the main conclusions and future works.

Cáceres P., Marcos E. and de Castro V. (2004).
INTEGRATING AGILE AND MODEL-DRIVEN PRACTICES IN A METHODOLOGICAL FRAMEWORK FOR THE WEB INFORMATION SYSTEMS DEVELOPMENT.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 523-526
DOI: 10.5220/0002637705230526

2 THE METHODOLOGICAL FRAMEWORK OF MIDAS

MIDAS is a model-driven methodological framework for agile development of WIS, which follows a service oriented approach. The MIDAS architecture (Cáceres et al., 2003) is based on the MDA of the OMG. MIDAS proposes: a) specifying the whole system by Computation Independent Models (CIMs), Platform Independent Models (PIMs) and Platform Specific Models (PSMs); and b) specifying the mapping rules between these models. Besides, MIDAS suggests using the Unified Model Language (UML) (OMG, 2003) as a unique notation to model the whole WIS.

In order to model the different aspects of the system it is important to identify them at the modeling level as are stated by Kulkarni and

Reddy (2003). To identify the aspects of a WIS, we have taken into account the middleware architectures of the Web service development platforms, as .NET or J2EE. So, we also propose a n-tier model architecture. Until now, our approach considers three aspects corresponding with the three tiers most commonly accepted: *graphic user interface*, *persistence* and *business logic* tiers. For the sake of uniformity with the Web Engineering terminology, these aspects will be called *hypertext*, *content* and *behavior* respectively (Retschitzegger and Schwinger, 2000). The advantage of this n-tier model architecture is that it is easily scalable; so, to incorporate new aspects, as security or management, we have just to introduce a new tier.

Summing up, MIDAS proposes to model the WIS according to two orthogonal dimensions (see figure 1): a) the MDA approach: that is, the degree of dependence of the platform; and b) the n-tier model architecture previously defined: that is, the aspects to be considered in a WIS. So, it defines PIMs and PSMs for each of the aspects above identified (hypertext, content and behavior), as well as the mapping rules between them.

3 TOWARDS A MODEL-DRIVEN WEB DEVELOPMENT IN AN AGILE WAY

As mentioned before, nowadays an interesting topic is the approach between agile and model-driven proposals (Turk et al., 2002). We also keep on considering agile practices. However, as is showed in figure 2, agile and model-driven process do not take into account the problem domain and the solution domain, in the same way (Weneger, 2002). Then our goal is the approach between these proposals, remarked as “area of interest” in figure 2. We focus on the analysis of their several aspects (Weneger, 2002), showed in table 2: People, process, technology, models and software.

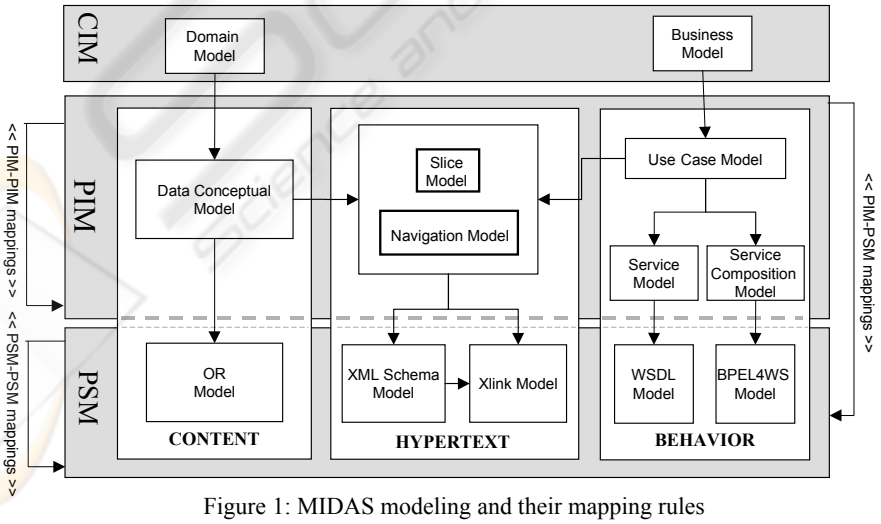

Figure 1: MIDAS modeling and their mapping rules

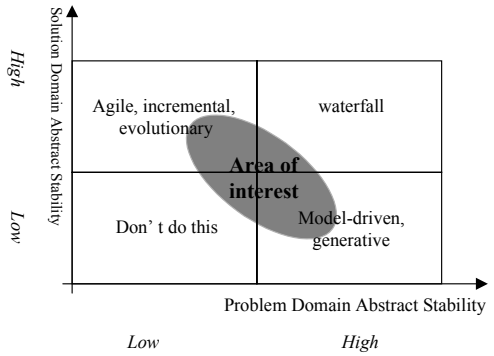

Figure 2: Relation between the problem and solution domain (Weneger, 2002)

Table 1: Distinguishing general aspects of agile and model-driven development process (Weneger, 2002)

Aspects	Agile	Model-driven
People	Have highest priority; interactions between customer and developer are facilitated.	No explicit role; problem domain model takes role of discussion platform for customer and developer.
Process	Has medium priority; ensures consistency of results of interactions between people; incremental, evolutionary.	No explicit role; strong tendency towards waterfall, less incremental processes.
Technology	Has lowest priority; only a means to an end; must be as simple as possible.	Is at center of approach; problem model (e.g., PIM) is manipulated, stored in and exchanged between repositories. Can be used to generate software (e.g., using a PSM).
Model	Secondary artifact; only produced up-front when absolutely need.	Primary artifact; source of generated implementation.
Software	Primary artifact; sole measure of progress	Secondary artifact; depending on solution domain, provides or adds aspects not covered by specification.

We study these different characteristics, and we adopt the most interesting practices for us. Next, we sum up the adopted characteristics in MIDAS:

People. As in agile processes, people have a high priority for us. We propose that the interaction between customer and the software project staff had priority over other interactions. In model-driven process, the customer only appears from the problem domain model. In our case, the customer and the software project start together from scratch and they also finish together. In this way, we can guarantee that the final product will satisfy to the customer.

Process. The process is an important aspect in our proposal. The MIDAS process is incremental, iterative, adaptive and with prototype, similar to the agile process and far from the waterfall model proposed in the model-driven processes. Also, we can obtain several software versions from an iterative and incremental process before the final version of software (Jacobson, 2000). Therefore, an agile and model-driven development can be also carried out in an incremental and iterative way, as is pointed by Ambler (2003).

Technology. Our proposal is addressed towards a specific technology (XML and object-relational). Then, unlike the agile process, we adopt different models from MDA (CIM, PIM and PSM) to represent platform independent and platform dependent models,.

Model. Modeling is also prioritary for us. The system knowledge must be written in a concise notation, as states Atkinson and Kühne (2003). In this way, we propose to use UML and UML profiles. The models and the code are the only documentation that we propose to generate. Then, in this point, we disagree with agile process.

Software. Our main goal is to satisfy the customer. We propose to generate a basic software version shortly. In this way, we propose to develop a software prototype in every iteration. We guarantee the software progress thanks both short development cycles and an incremental process. In this aspect, we agree with agile process.

4 CONCLUSIONS AND FUTURE WORKS

In this paper, we focus on the integration of model-driven and agile practices, adopting the most interesting practices in the MIDAS methodological framework for WIS development. MIDAS combines the MDA and a three- layer architecture which represent different WIS views (hypertext, content and functionality). Based on this architecture, MIDAS proposes system modeling with CIMs, PIMs and PSMs and defines mappings between the mentioned models.

Now, MIDAS is been validating by the CASE tool development, named *MIDAStool*. Our aim is the model-driven and agile practices integration into a methodological environment for semi-automating WIS development.

REFERENCES

Ambler, S. (2003). Agile Model Driven Development is Good Enough. *IEEE Software*, Vol. 4, pp. 71-73, september-october de 2003.

Atkinson, C., Kühne, T. Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, Vol. 4, pp. 36-41, september-october de 2003.

Cáceres, P.; Marcos, E. (2001) Agile Process for Web development. *Web Engineering Workshop in Working days of Software Engineering and Databases, 2001. (JISBD2001).* Ciudad Real (Spain). In http://www.dlsi.ua.es/webe01/articulos/s112.pdf (in spanish).

Cáceres, P.; Marcos, E.; Vela, B. A MDA-Based Approach for Web Information System Development. *Workshop in Software Model Engineering (WiSME@UML'2003) in conjunction with UML Conference.* October, 2003. San Francisco, USA. Accepted.

Fowler, M. (2001). The New Methodology. Retrieved May 2001 from http://www.martinfowler.com/articles/newMethodology.html.

Fraternali, P. (2000). Tools and Approaches for Developing Data-Intensive Web Applications: a Survey. Retrieved July 2000 from the World Wide Web: http://toriisoft.com

Jacobson, Booch and Rumbaugh. *The Unified Software Development Process*. Addison Wesley, 1999.

Kulkarni, V., Reddy, S. (2003). Separation of Concerns in Model-Driven Development. *IEEE Software*, Vol. 4, pp. 64-69, september-october de 2003.

Lowe, D.; Hall, W. (1999). Hypermedia & the Web. An Engineering Approach. J. Wiley and Sons.

Marcos E., Vela B., Cavero J. M. (2001a) Extending UML for Object-Relational Database Design. Fourth Int. Conference on the Unified Modelling Language, UML 2001, Toronto (Canadá), LNCS 2185, Springer Verlag, pp. 225-239, 2001.

Marcos, E., Vela, B., Cavero J.M., Cáceres, P. (2001b) Aggregation and Composition in Object-Relational Database Design. 5 th. Conference on Advances in Databases and Information Systems. Ed.: Albertas Caplinskas and Johann Eder, pp.195-209, 2001.

Marcos, E. Cáceres, P., Vela, B., Cavero, J.M. (2002) MIDAS/DB: a Methodological Framework for Web Database Design. DASWIS 2001. Yokohama (Japan), November, 2001. LNCS-2465. Springer Verlag. ISBN 3-540-44122-0. September, 2002.

Marcos, E. , Vela, B. and Cavero J.M. (2003a) Methodological Approach for Object-Relational Database Design using UML. Journal on Software and Systems Modeling (SoSyM). Springer-Verlag. Ed.: R. France and B. Rumpe. ISSN: 1619-1366. Volumen SoSyM 2, pp.59-72, 2003.

Marcos, E., De Castro, V., Vela, B. (2003b) Representing Web Services with UML: A Case Study. International Conference on Service Oriented Computing. Trento (Italia). 15-18 december 2003. Accepted.

Miller, J. and Mukerji, J. (Eds). (2001) MDA Guide Version 1.0. Document number omg/2003-05-01. Retrieved from: http://www.omg.com/mda, 2003.

OMG. OMG Unified Modeling Language Specification. Version 1.5. Retrieved from: http://www.omg.org/technology/documents/formal/uml.htm, 2003.

Retschitzegger, W., Schwinger W. Towards Modeling of Data Web Applications- A Requirement's Perspective. In Proceedings of the America's Conference on Information Systems. V.I, pp. 149-155, 2000.

Turk, D., France, R., Rumpe, B., Georg, G. Model-Driven Approach to Software Development. Advances in Object-Oriented Information Systems. LNCS 2426. Springer-Verlag, ISBN: 3-540-44088-7, pp.229-230, September, 2002.

Vela, B. and Marcos E. (2003a). Extending UML to represent XML Schemas. 15th Conference On Advanced Information Systems Engineering (CAISE '03). Klagenfurt/Velden (Austria). 16-20 June 2003. Ed. J. Eder, T. Welzar. ISBN: 86-435-0549-8. 2003.

Vela, B. and Marcos E. (2003b) Web Information Systems Development: From navigational conceptual modeling to logic modeling in XML. In Working Days on Software Engineering and Databases. Alicante, Spain. JISBD 2003 (in spanish).

Weneger, H. (2002). Agility in Model-Driven Software Development? Implications for Organization, Process and Architecture. Retrieved from http://www.softmetaware.com/oopsla2002/wenegerh.pdf