Integrating an Agile Process in a Model Driven Architecture

Paloma Cáceres, Francisco Díaz, Esperanza Marcos

Rey Juan Carlos University
C/ Tulipán s/n, 28933-Madrid (SPAIN)
pcaceres@escet.urjc.es
fjdiaz@escet.urjc.es
emarcos@ escet.urjc.es

Abstract. The model driven development is an interested area among software engineers as well as the agile development. In fact, combining model driven and agile practices is an interesting perspective for the software engineering. In this work, we present a case study of *agile model driven development* within the MIDAS framework, which composes a model driven architecture based on the Model Driven Architecture (MDA) with a set of agile practices and activities based on eXtreme Programming (XP). The case of study has been carry out in the Rey Juan Carlos University, with students of a Software Engineering subject. MIDAS has been refined applying the learned lessons from the results of this case study.

1 Introduction

In this paper1 we present a case study of agile model driven development (AMDD) [Am03] within the framework of MIDAS [Ma01]. MIDAS is a model driven methodology for Web Information Systems (WIS) agile development.

Initially, to combine the advantages of the agile practices with the advantages of the model driven ones [Am03],[Tu02],[We02], we consider the possibility of integrating both kind of practices in our MIDAS methodological framework [CMC04]. In this way, our methodology is based on: a) a model driven architecture [CMV03], based on the

[1] This research is carried out in the projects framework: EDAD (07T/0056/2003 1) financed by Autonomous Community of Madrid and DAWIS, financed in part by the Ministry of Science and Technology of Spain (TIC 2002-04050-C02-01).

Model Driven Architecture (MDA) [Om01] proposed by the Object Management Group (OMG), and b) a set of agile practices and activities mainly based on XP [Be01]. The MIDAS architecture is based on the MDA and also takes into the separation of aspects. MDA address the development through models and mapping rules between models [Om01]. The separation of aspects at the modeling level is important in order to model the different aspects of the system independently, as it is stated in [KR03]. In this way, the model driven development proposed in MIDAS facilitates the development, taking into account the separation of aspects both at modeling and programming level; and also guides the development through the models and their mapping rules.

As a result of this work, in this paper, we propose an *agile and model driven development*, which has carried out by means a case study. It was realized in the Rey Juan Carlos University (URJC) of Madrid (Spain) with students of the subject 'Software Engineering II' from the last course of the Computer Science Engineering degree. The developers described the experience as positive and valuable, emphasizing the identification of different views of the system according to the *separation of aspects* proposed in MIDAS.

The rest of the paper is structured as follows: section 2 is an overview of MIDAS architecture. In section 3 the case study is presented. Finally, in section 4, we conclude underlying the main contributions and the future works.

2 The Model driven Architecture of MIDAS

The MIDAS architecture combines a model driven architecture with an n-tier architecture. On one hand, the MIDAS architecture is based on MDA and specifies: a) the whole system by means of Computation Independent Models (CIMs) which represent the problem domain, and by Platform Independent Models (PIMs) and Platform Specific Models (PSMs), both representing the solution domain; b) the mapping rules between models. On the other hand, MIDAS is based on a n-tier architecture because, as it is stated in [Am03], at the modeling level it is important to identify the different aspects of the system in order to model them independently. To identificate the aspects of a WIS we have taken into account the middleware architectures of the Web service development platforms, as for example, .NET or J2EE, because MIDAS follows a service-oriented approach [MCCb04]. Until now, our proposal considers three aspects corresponding with the three tiers most commonly accepted: *graphic user interface*, *persistence* and *business logic*. For the sake of uniformity with the Web Engineering terminology, these aspects will be called from now on *hypertext, content* and *behavior,* respectively [RS00]. One of the advantages of this n-tier model architecture is that it is easily scalable and, so, to incorporate new aspects, as security or management, we just have to add a new tier.

Summing up, the MIDAS architecture proposes to model a WIS according to two orthogonal dimensions (see figure 1): a) the degree of dependence to the platform, according to the MDA approach; 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.

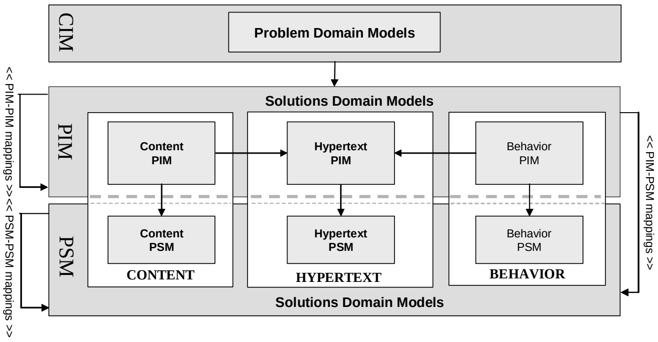

Fig. 1. Model driven architecture of MIDAS

The MIDAS architecture is presented in detail in [CMV03]. Other works related to MIDAS, as for example the UML extension for object-relational database, XML schema and WSDL models, which can be found in [MCV03],[MVC03],[VM03]; [Ma01] define the development process of the content aspect; finally, [MCC04],[Ca04] describe the MIDAS method for modeling the hypertext aspect of a WIS, at the platform independent level.

3 The Case Study

3.1 The Framework

The experience has taken place in the Rey Juan Carlos University (URJC) of Madrid (Spain) with students of the subject Software Engineering II, of the fifth and last course of the Computer Science Engineering degree. Eight groups of four to ten students have been constituted. The experience degree of the programmers and analysts has varied between low and medium. Two teachers have been involved in the experience. One has played the customer role and has worked in the classroom with the students 4 hours a week. The rest of the time the customer is also available in Internet. The other teacher has played the manager role, requiring the development execution.

The teams have been divided in pairs. Every pair in a team has had to develop the same user stories, generating the engineering tasks, the release planning, and the iterations and implementations of each product. In addition, every pair has had to pass the unit tests to their own software. Unlike XP, the pairs have been the same along the whole experience.

The development has been carried out according to the four iterations from MIDAS, generating one software release. All artifacts proposed in MIDAS have been generated and have always been available in http://www.escet.urjc.es/~aisw/.

The set of activities has been instanced the following way: Release Planning of version, Architecture and implementation requirements Selection, Iterations Planning according to previous selection, Developed tests, Web page Implementation and User validation of the product’s previous iteration.

Tests have always been implemented by a different team that the one who developed the software product. So, a pair from the A team has developed the A software, but has tested the results from the software generated by one of the pairs from the B team. Every pair has had to test their own products and the products of another different pair.

3.2 MIDAS process

The MIDAS process is based on a set of practices which applied to the model driven architecture described above, makes agile and model driven the Web development, facilitating the iterative, incremental and prototyping-based development. MIDAS AMDD helps to generate the code and the models in an iterative and incremental way. As Ambler states in [0], "*Notably with AMDD, programmers write the code progressively in step with the models, in which implementation occurs iteratively and incrementally.*

Before beginning a software development project, the system context has to be identified (domain and business model). In this way, MIDAS proposes that the domain model is represented by the class diagram, and the business model is represented by the use cases diagram.

The MIDAS proposes four different work flows: *requirements*, *hypertext*, *content* and *behavior*. The goal of the *requirements* workflow is to obtain the user and system requirements. For the development of the rest of work flows, the necessary models must be obtained, and then the transformation guidelines have to be applied to finally obtain the rest of the models. The details about the hypertext, content and behavior modeling are described in [CMV03],[MCC04]. MIDAS proposes using the UML as a unique notation to model CIM, PIMs and PSMs. Prototypes and implementation are realized at the moment they are needed.

The goal of each work flow is the development of a software product formed by the products obtained after executing the following activities: Release Planning, Architecture and requirements selection, Iteration Planning according to previous selection, Modeling, Writing the unit tests, Implementation, Testing.

We want to emphasize that the set of activities will be instanced in every work flow, generating their specific software products associated because one work flow is different to another. Also, one release is different to another. In this way, MIDAS defines a generic process which will be instanced when executed.

3.3 Lessons Learned

To finish the case study, we made a set of questions to the students focused on collecting their opinions about MIDAS. In short, some lessons learned from this experience are:

- It is very helpful to offer a first general vision of the system to develop, and that can be achieved during the release planning, through a first user history. The business and domain models help to show the knowledge graphically and facilitates the incorporation of any new member into the team. This has been demonstrated by the evaluation of a pair by another pair, using tests. This case study has been very positive for the team, which has underlined the usefulness of having a common vocabulary from the start, and also a first approach of the range and importance of the project.

- The principle of aspects separation from MIDAS has been very useful for the teams, facilitating the requirements agrupment and analysis, because it helps a lot in the prioritization and planning of user stories in the first iteration.

- Development's standards. The use of standards (for example, UML), helped the communication, and the feeling of collective ownership. In addition, each pair established its own programming standard due to the time restriction.

- Pair Development. This practice has been named as one of the most positives by developers, who felt higher security and greater responsibility and commitment during all the process, thanks to this practice.

- Continuous Integration. Having the software available via Web, forces to maintain the consistency of all the information generated by the project. The developers indicated that this maintenance is expensive. But they also emphasize its importance, since they had to test the software of a different pair.

4 Conclusions and Future Works

In this work we present a case study of WIS development integrating XP and the MIDAS model driven architecture within the methodological framework of MIDAS.

The integration process has been made in several stages. First, we study the main characteristics of the model driven and agile practices. Afterwards, we analyze the possibility of integration of both practices, and then, they have been modified and adapted according to the development process of MIDAS. Finally, we are testing the agile model driven development process obtained. The experience of working with MIDAS has been satisfactory and positive for the developers. The main advantages of the agile and model driven development of MIDAS are: To strengthen the knowledge about the final goal of agile modeling; To guide the developers using the indicated models and their mapping rules; To work based on the separation of aspects; Additionally, in MIDAS some important issues of agile and model driven development have been checked, as the possibility of modeling and coding simultaneously.

The future works are focused on a CASE tool based on the MIDAS architecture for the semi-automatic code generation, based on the proposed standards for development.

References

[Am03] Ambler, S.W.: Agile Model Driven Development is Good Enough. IEEE Software, pp. 71-73, September-October, 2003.

[Be01] Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley Ed. 2001.

[Ca04] Cáceres, P., Marcos, E., de Castro, V.: Navigation Modeling From A User Service Oriented Approach. In the International Conference on Advances in Information Systems. ADVIS'04. Izmir (Turkey). Accepted

[CMC04] Cáceres, P., Marcos, E., de Castro, V.: Integrating Agile and Model Driven Practices in a Methodological Framework for the Web Information Systems Development. In the 6th International Conference on Enterprise Information (ICEIS 2004).

[CMV03] Cáceres, P., Marcos, E., Vela, B.: A MDA-Based Approach for Web Information System Development. Software Model Engineering Workshop in UML'2003.

[KR03] Kulkarni, V., Reddy, S.: Separation of Concerns in Model driven Development. IEEE Software, Vol. 4, pp. 64-69, September-October, 2003.

[Ma01] Marcos, E., Cáceres, P., Vela, B., Cavero, J.M.: MIDAS/DB: a Methodological Framework for Web Database Design.Springer Verlag. ISBN 3-540-44122-0. September, 2002.

[MCC04] Marcos, E., Cáceres, P., De Castro, V.: An approach for Navigation Model Construction from the Use Cases Model. The 16th Conference On Advanced Information Systems Engineering. CAISE’04 FORUM, pp. 83-92.

[MCV03] Marcos,E., De Castro,V., Vela,B.:Representing Web Services with UML: A Case Study. Conference on Service Oriented Computing (ICSOC’03). Springer Verlag, pp. 17-27, 2003.

[MVC03] Marcos, E., Vela, B., Cavero, J.M.: Methodological Approach for Object-Relational Database Design using UML. Journal on Software and Systems Modeling (SoSyM). Springer-Verlag. France, R., Rumpe, B. (eds.) Volume SoSyM 2, pp.59-72, 2003.

[Om01] OMG. OMG Model Driven Architecture. Miller, J., Mukerji, J. (eds.) 2001. Document number ormsc/2001-07-01. Retrieved 2003, from: http://www.omg.com/mda.

[RS00] Retschitzegger, W., Schwinger, W.: Towards Modeling of Data Web Applications-A Requirement’s Perspective. Proceedings of the America’s Conference on Information Systems, pp. 149-155, 2000.

[Tu02] Turk,D., France,R., Rumpe,B., Georg,G.:Model driven Approach to Software Development. Advances in Object-Oriented Information Systems. Springer Verlag, pp.229-230, 2002.

[VM03] Vela,B., Marcos,E.: Extending UML to represent XML Schemas. Conference On Advanced Information Systems Engineering. CAISE’03 FORUM. J. Eder, T. Welzer (eds.). 2003

[We02] Weneger, H. (2002). Agility in Model driven Software Development? Implications for Organization, Process and Architecture. Retrieved from http://www.softmetaware.com/oopsla2002/wenegerh.pdf