Model-driven in reverse. The practical experience of the AQUA project

M.J. Escalona, J.J. Gutiérrez
University of Seville. ETS. Ingeniería Informatica. Av. Reina Mercedes S/N. Seville, Spain
mjescalona@us.es, javierj@us.es

L. Rodríguez-Catalán
Emasesa.
Escuelas Pías nº1. Seville, Spain
lrodriguez@emasesa.com

A. Guevara
University of Málaga. ETS. Ingeniería Informatica. Campus de Teatinos S/N. Málaga, Spain
guevara@uma.es

ABSTRACT

Model-Driven Web Engineering (MDWE) is a new paradigm which provides satisfactory results in software development. However, most experiments with MDWE towards the development of software from the requirements to the implementation stages. However, in practice, sometimes software development does not follow this classic path. In this paper, MDWE is presented as a suitable selection in software projects of technological evolution. The AQUA project is a very large and important project which is a fusion of three previous systems. Thus, this project starts with the original code of these three systems. The paper introduces the power of MDWE in this kind of systems.

Categories and Subject Descriptors

D.2 [**Software Engineering**]: Requirements / Specifications

General Terms

Documentation, Design, Standardization, MDWE

Keywords

Model-Driven Engineering. Web Quality. Web Engineering.

1. INTRODUCTION

Model-Driven Web Engineering is an effective paradigm which, in recent years, has been by several web approaches: UWE (UML Web Engineering)[15] and WebML [5] are examples.

In MDWE, MDA (Model-Driven Architecture)[20], the standard Model-Driven environment defined by OMG, has been adapted as presented Figure 1, obtained from [17]. Three set of models may be identified from figure 3: CIM, PM and PM, which are described in next paragraphs.

In the CIM (Computer Independent Model) level, models oriented towards capturing business logic are defined, for instance, requirements models. Specific models for the Web environment such as navigational and presentation models can then be obtained by means of a set of CIM-to-PIM transformations.

EATIS'09, June 3–5, 2009, Prague, CZ.

PIM to PSM (Platform Specific Model) transformations enable models specific for the development platform, for instance J2EE and .NET, to be established.

Finally, PSM-to-code transformations provide the implementation of the system.

This is the logical path in system development. Starting from requirements and by using a set of transformations, the final code is obtained. MDWE offers important advantages in Web development.

It enables system traceability and improves development because it offers a guide to development and reduces the development time.

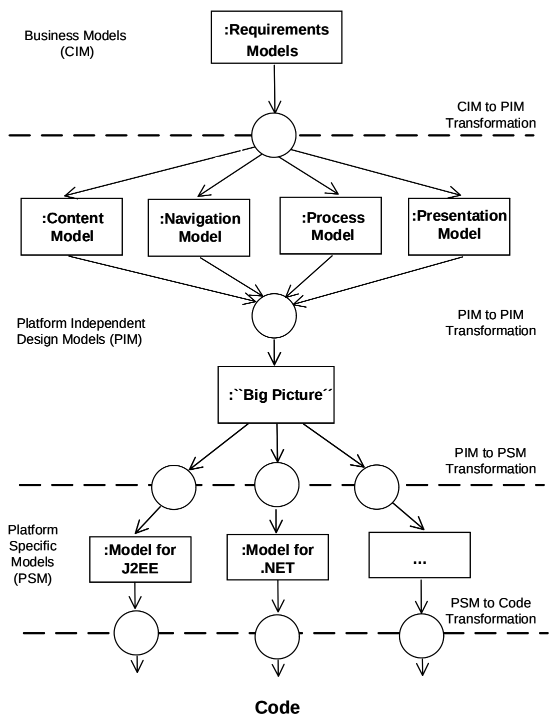

Fig. 1. Model-Driven Web Engineering

However, what happens when requirements are already defined?. In practice, software can sometimes be found that starts from old code. These systems have no previous documentation, the requirements are not written and the starting point of the system is the old code that has to be migrated, or event when different subsystems are written using different languages and platforms. Is MDWE a suitable selection in these cases? Can it offer special advantages in the development of these systems? The aim of this paper is to solve these questions.

This paper introduces a practical real project in which these questions were solved. In Section 2, the paper presents NDT (Navigational Development Techniques)[13]; a MDWE methodology which has been applied in the AQUA project presented in Section 3. The AQUA project is a large system of technological migration that fuses three different systems: AQUA-Red, AQUA-SIGO and AQUA-SIG. In Section 4, the paper presents advantages offered by MDWE in the project. The paper finishes with related work and conclusions.

2. Navigational Development Techniques (NDT)

NDT is a methodological Model-Driven Web process focused on the requirements and the analysis phases. It offers a systematic way to deal with the special characteristics of the web environment. NDT is based on the definition of formal metamodels, presented in [13], that allows the creation of derivation relations between models. NDT takes this theoretic base and complements it with elements necessary to define a methodology, (techniques, models, methodological process, etc.) in order to offer a suitable context for the application to real projects.

Following these lines, NDT starts with the theoretic definition of the requirements engineering metamodels and proposes a methodological environment to lead the team in the capture, definition and validation of requirements. With the theoretic base of metamodels and relations, the next phase is the analysis phase, whereby three models are generated. These models are:

1. The conceptual model, which defines the static structure of the information and its relations.
2. The navigational model, which defines how users can navigate through the information.
3. The abstract interface model, which is composed of a group of HTML and XML prototypes that enable the conceptual and navigational models.

However, the generation of these three models is carried out in two phases. In the first, analysis models are generated systematically from the requirements using CIM-to-PIM transformations defined between models. These models are called basic analysis models. In these basic models, the analyst can make some changes in order to render them more suitable and thereby obtains the final analysis models.

NDT also normalizes the structure of the results that must be developed during the requirements engineering and the analysis phase by offering a complete definition of each document obtained.

In conclusion, NDT can be defined as a methodology for the requirements and analysis phases, the rest of the life cycle is dealt with other approaches. NDT is offered to cover a gap in the treatment of the first phases of the life cycle in Web Engineering.

One of the most important advantages of NDT is its practical application to real projects and in the business environment. NDT was extended in order to cover the whole life cycle, described in [11].

In order to support the development of NDT, a set of tools were developed. This set of tools was named NDT-Suite and is composed by four tools:

1- **NDT-Profile**: This is a specific profile for NDT, developed using Enterprise Architect[10]. This tool offers an environment to define specific profiles and NDT-Profile has adapted Enterprise Architect to support each artifact of NDT.

2- **NDT-Driver**: This is a tool to execute transformations of NDT. NDT-Driver is a Java-free tool that implements QVT Transformations of NDT and allows obtain analysis models to be obtained automatically from the requirements model.

3- **NDT-Quality**: This is a tool that automatically checks the quality of a project developed with NDT-Profile.

4- **NDT-Report**: This is a tool that prepares formal documents which are then validated by final users and clients. For instance, it enables the automatic generation of a Requirements Document with the format defined by clients.

All these tools and their manuals can be downloaded from www.iwt2.org. In Figure 2, the life cycle of the extended version of NDT and tool support is presented.

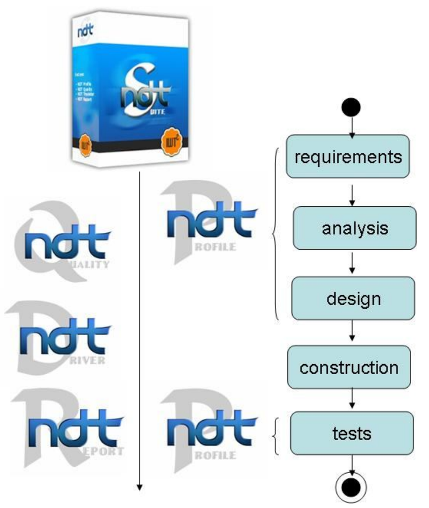

Fig. 2. NDT extended life cycle and NDT-Suite support.

3. The AQUA Project

Emasesa, (Empresa metropolitana de abastecimiento y saneamiento de aguas de Sevilla, S.A.), has been engaging in complete urban water cycle management, supplying water for over one million users, some in heavinly industrialized areas, for over thirty years.

Emasesa has to ensure the water supply to thousands of people. It has to ensure the quality of this supply, solve problems in the supply and control the correct use of water in Seville.

The AQUA project is a necessity of Emasesa for the integration of its previous systems: client management system (AQUA-SiC), Network management system (AQUA-ReD) and Project and infrastructure management systems (AQUA-SigO), into a single system and modernize the technological platform of the systems.

The project has four main objectives:

1. To migrate AQUA-SiC, AQUA-ReD and AQUA-SigO into a common Java J2EE platform and to unify them into a single system: AQUA. This is AQUA core.
2. To develop an integrated system to support the decision management in AQUA. This is DSS system.
3. To offer a suitable way based on an integration bus to connect AQUA with other transversal systems in Emasesa.
4. To implement a friendly interface with a suitable connection for both clients and employments. This is FrontEnd.

In Figure 3, the architecture of AQUA is presented.

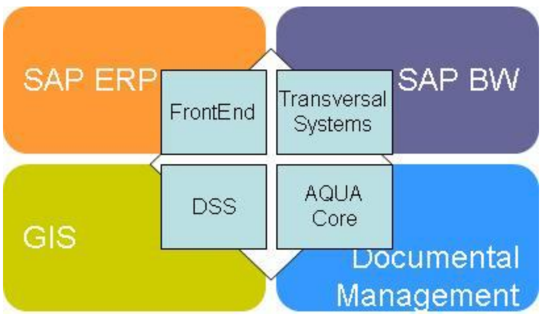

Fig. 3. AQUA Architecture

AQUA Core, FrontEnd, DSS and the communication with transversal systems are based on the use of SAP ERP and SAP BW, in a Geographical Information System (GIS) and on a system to manage documental resources in the project.

The development of AQUA is fronted by a mixed group composed by two companies: Capgemini[4] and Novasoft[19] and two universities: the university of Málaga and the university of Seville.

4. MDWE in Reverse

The development process of AQUA is based on an iterative life cycle that follows the Unified Process approach[21].

Table 1: TableToClass transformation

```
Transformation TableToClass (db : DB, ndt :NDT){
 top relation R1{
   checkonly domain db t:Table  {
     name = n,
    description = d
    };
   enforce domain ndt cl:Class{
     name = n, description = d
    };
 }
 relation R2{
   checkonly domain db tbAt : Attribute  {
     namespace = tb: Table{},
     name =tb
   };
   enforce ndt cat: Attribute {
     namespace = cc: Class{},
     name = tn
   }
  when {R1(tb, cc);
 }
}
```

During this section, we are going to analyze how AQUA is being developed and how MDWE is helping to manage the quality of the technological migration.

STEP 1.- Database Migration: The first step was the database migration. The new database management system is Oracle. In the migration of the database, the original database was reviewed and corrected. A special artifact in NDT-Profile was defined not only to define but also to capture tables and attributes. This artifact includes some fields to name and describe each table and each attribute in these tables, and it also includes a special section to detail some functional aspects (triggers or PL/SQL sentences) that were included in the original database. NDT-Quality was used in order to check that all the database artifacts in NDT-Profile were correctly defined.

STEP 2.- Code Analysis: At the same time as database migration, another part of the development team analyzed the code and defined their functional requirements. This activity is a little different to other developments since clients were not interviewed for requirements capture, these requirements were obtained from code. NDT-Quality was also useful in the quality control of this step. With the option to check requirements, this tool enabled errors and inconsistencies in the definition of use cases to be identified.

STEP 3.- Derivation of Classes: Using QVT transformations, analysis models were defined from the requirements. In order to trace the correspondence between requirements and analysis, some traceability matrixes were defined into NDT-Profile. With NDT-Quality, the traceability between requirements and analysis is automatically analyzed. Thus, traceability is ensured.

STEP 4.- Traceability between tables and classes: Again using QVT but in reverse, and by using NDT-Quality, traceability between tables and persistence classes were tested. Tables were derived from the original database and classes were systematically derived from code. Each class has to be related with one or more tables and, vice versa. Attributes in tables and classes have to be the same; at least, attributes in classes must be in tables.

Thus QVT transformations and checks based on these transformations are used in two ways: from requirements to classes and from tables to classes.

5. A Basic Example

In this section, this unified process is going to be illustrated with a simple example based on infrastructure work information.

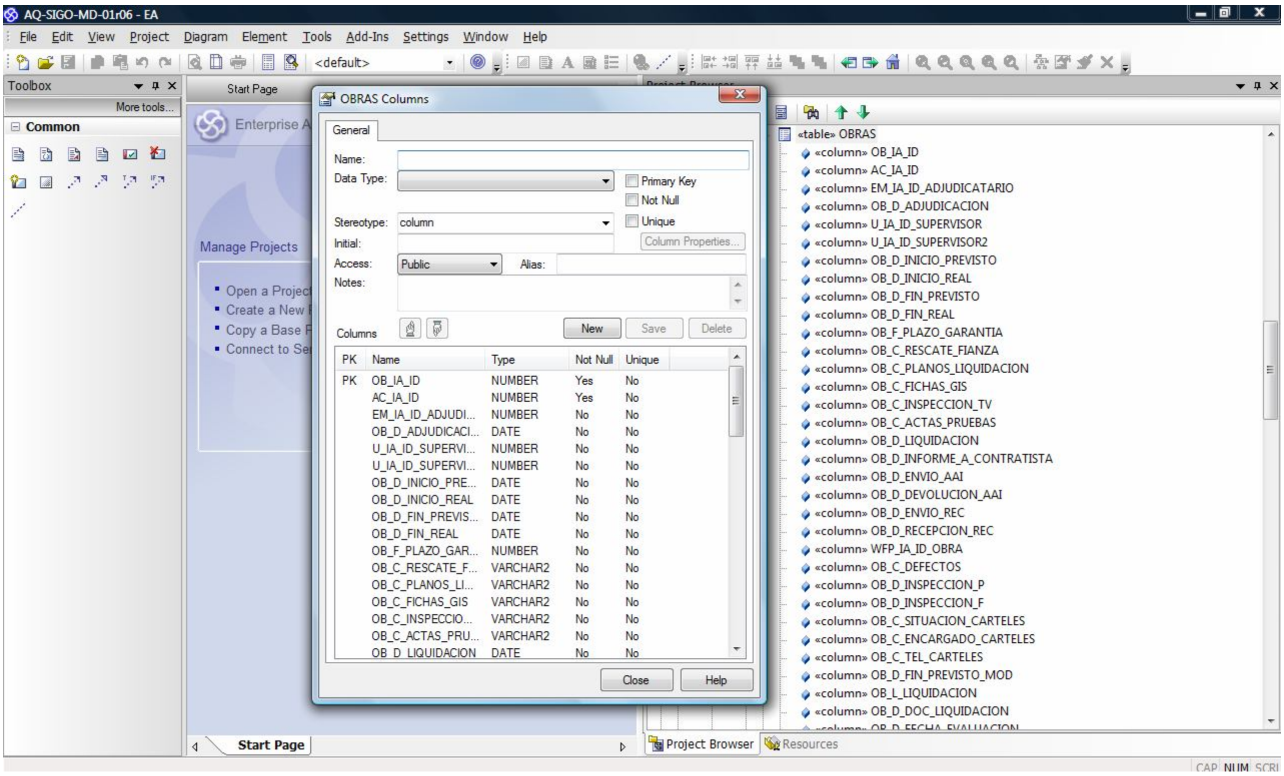

Fig. 4. NDT-Profile for AQUA.

The management of distribution/production infrastructure necessary for water supply is a very important part of AQUA. Originally, management was a part of the database that stored all information about these infrastructure works[1].

In Figure 4 a capture of the interface of NDT-Profile for this artifact is presented. As it can be analyzed, any attribute was defined.

From this table, a class in the design phase has to be designed with similar attributes.

To obtain this class, a QVT Transformation has to be executed. This transformation is represented in Table 1.

Basically, each table generates a class with the same attributes.

This transformation can be executed automatically with a special version of NDT-Driver developed for AQUA project.

Transformations to obtain analysis models from requirements in NDT can be observed in [13].

From this process, we start with a set of requirements, and classes are obtained.

Both products (classes from tables and classes from requirements) have to coincide. NDT-Quality was a power tool for the automatic check of QVT-Transformations and these constraints.

Another important advantage of this environment is the use of only one tool to manage any information of any lifecycle phase. Enterprise Architecture and the adaptation of NDT-Profile makes up every artifact of the project. For this reason NDT-Quality therefore verifies all the rules over NDT-Profile.

All developers in AQUA introduce artifacts in this file, so traceability and uniformity is ensures using Enterprise Architect and NDT-Profiles utilities.

As can be deduced from this section, MDWE is useful in this project since it enables traceability in either direction: from the requirements to code and from the code to the analysis.

Emasesa has the certainty that its code is traced into the new application using a systematic method.

6. Related Work

Nowadays, MDE and MDA are very often applied and this is also true in the Web Engineering environment. This paper is mainly focused on the inverse way. How can MDWE be useful to check traceability in both directions? Although, work oriented in this sense were not found in the bibliography, some related work can be referenced.

One of the most recent paper is [24]. This paper presents an approach for the transformation of a Web requirements model into a set of prototypes. A requirements treatment based on the task metaphor is proposed. Valderas et al. offer an extension of this approach which deals with the specific characteristics of Web requirements. A way to derive the navigational model of OOWS[14] is then presented. Firstly, they propose a definition of requirements as tasks; these tasks are translated into an AGG Graph. Using Graph transformations, analysis models are obtained. The approach is supported by a readily available tool, thereby providing both a suitable solution and support. However, its transformations are not based on OMG tendencies which leads to incompatibility with other similar approaches.

In [11], the power of metamodels is presented. In comparative studies on Web approaches, a general conclusion is that similar concepts are used or represented with a different number of models, techniques or artifacts. Thus, for instance, navigational classes are presented with different elements in UWE, OOHDM or WebML. Escalona and Koch show in this paper how metamodels can represent concepts independently of their representation and notation, and how only concepts are important. A metamodel for Web requirements, named WebRe is given, which represents requirements models of W2000 [2], NDT, OOHDM and UWE. In [17], their work is extended using QVT to obtain analysis models from this metamodel. These papers are interesting since they are completely based on UML and QVT: standards defined by OMG. However, the results are too theoretical and lack application.

Another interesting case is WebML. In [23][18], some metamodels for WebML can be found. Metamodels are shown in the representation of classic concepts independent of the artifact used to represent them.

7. Conclusions

This paper presents the power of MDWE in the traceability in Software development. MDWE is a new paradigm that offers satisfactory results in Web development.

This paper shows, based on a practical real case, how MDWE can offer other ways to improve web software development and to measure the quality of the system during its development.

The AQUA project is currently under development, using an iterative process. To date, only the first iteration has been developed. The process presented in Section 4 is being applied in every iteration and our tools, metrics and quality measures are being continually improved.

AQUA must be finished by the beginning of 2010. Every fifteen days a strategy committee has a special meeting to follow the project and to analyze metrics that are defined to measure the project.

As a research aim, AQUA has opened an important line of research to be continued. The use of MDWE to assure traceability is an important advantage that could offer suitable results for the Web Engineering community. The adaptation of our approach, NDT and NDT-Suite, to this kind of project, oriented towards the improvement of this project constitutes a major challenge.

[1] In Spanish, this table is named “obras”. This is the name that appears in the figure.

8. ACKNOWLEDGMENTS

This research has been supported by the project QSimTest (TIN2007-67843-C06_03) and by the RePRIS project of the Ministerio de Educación y Ciencia (TIN2007-30391-E), Spain.

9. REFERENCES

[1] ArgoUWE. http://www.pst.informatik.uni-muenchen.de /projekte/argouwe

[2] L. Baresi, F. Garzotto, M. Maritati.. "W2000 as a MOF Metamodel". 6th World Multiconf. on Systemics, Cybernetics Informatics. 2002

[3] C. Cachero. "Una extensión a los métodos OO para el modelado y generación automática de interfaces hipermediales". PhD Thesis. University of Alicante. Alicante, Sapin. 2003.

[4] Capgemini. www.es.capgemini.com

[5] S. Ceri, P. Fraternali, P. Bongio. "Web Modelling Language (WebML): A Modelling Language for Designing Web Sites". Conference WWW9/Computer Networks 33 (1-6) pp. 137-157. Mayo 2000.

[6] Consjería de Cultura. Junta de Andalucía. www.juntadeandalucia.es/ccul.

[7] Y. Deshpande, S. Marugesan, A. Ginige, S. Hanse, D. Schawabe, M. Gaedke, B. White. "Web Engineering". Journal of Web Engineering. Vol. 1 Nº 1. pp. 3-17.2002. Rinton Press.

[8] A. Durán, B. Bernárdez, A. Ruiz, M. Toro. A Requirements Elicitation Approach Based in Templates and Patterns. Workshop de Engenharia de Requisitos. Buenos Aires, Argentina. 1999.

[9] Emasesa. Empresa Municipal de Aguas de Sevilla. http://www.aguasdesevilla.com

[10] Enterprise Architect. www.sparxsystems.com

[11] M.J. Escalona, N. Koch. "Metamodelling the requirements of Web Systems" Proceedings of the 2nd International Conference on Web Information Systems and Technologies (WebIST 2006). pp.310-317. 2006.

[12] M.J. Escalona, G. Aragón, J.J.Gutierrez, J.A. Ortega, I. Ramos. NDT & Metrica v3. An approach for public organization in Spain International Conference on Web Information Systems and Technologies (WebIST'08). Madeira, Portugal. To be published. May, 2008

[13] M.J. Escalona, G. Aragón. NDT NDT. A Model-Driven approach for Web requirements. IEEE Transaction on Software Engineering. Vol. 34. Nº3. pp.370-390. 2008.

[14] J. Fons, V. Pelechano, M. Albert, O. Pastor. "Development of Web Applications from Web Enhanced Conceptual Schemas". Conference on Conceptual Modeling LNCS, 2813, pp. 232-245. Springer Verlag 2003

[15] N. Koch. "Software Engineering for Adaptive Hypermedia Applications". Ph. Thesis, FAST Reihe Softwaretechnik Vol(12), Uni-Druck Publishing Company, Munich. Germany. 2001.

[16] N. Koch. "Transformations Techniques in the Model-Driven Development Process of UWE." 2nd Model-Driven Web Engineering Workshop (MDWE 2006), ACM Vol. 155. 2006.

[17] N. Koch, G. Zhang, M.J. Escalona. "Model Transformations from Requirements to Web System Design". ACM International Conference Proceeding Series. Proceedings of the 6th international conference on Web Engineering (ICWE 2006). Ed. ACM. pp. 281-288. 2006.

[18] N. Moreno, P. Fraternali, A. Vallecillo. "A UML 2.0 Profile for WebML Modelling". II International Workshop on Model-Driven Web Engineering. Palo Alto, California. 2006.

[19] *Novasoft*. www.novasoft.es

[20] OMG: MDA Guide, http://www.omg.org/docs/omg/03-06-01.pdf. Version 1.0.1. 2003.

[21] OMG. Unified Modeling Language: Superstructure, version 2.0. Specification, OMG, 2005. http://www.omg.org/cgi-bin/doc?formal/05-07-04.

[22] G. Rossi. "An Object-Oriented Method for Designing Hypermedia Applications". PHD Thesis. University of PUC-Rio. Rio de Janeiro. Brazil, 1996.

[23] A. Schauerhuber, M. Wimmer, E. Kapsammer. "Bridging existing Web Modeling Languages to Model-Driven Engineering: A Metamodel for WebML". II International Workshop on Model-Driven Web Engineering. Palo Alto, California. 2006.

[24] P. Valderas, V. Pelechano, O. Pastor. "A Transformational Approach to Produce Web Application Prototypes from a Web Requirements Model". International Journal of Web Engineering and Technology (IJWET) (1476-1289). 2006.

[25] WebRatio. http://www.webratio.com