

# Developing eBusiness Solutions with a Model Driven Approach: The Case of Acer EMEA

Roberto Acerbis<sup>1</sup>, Aldo Bongio<sup>1</sup>, Marco Brambilla<sup>2</sup>,  
Massimo Tisi<sup>2</sup>, Stefano Ceri<sup>2</sup>, and Emanuele Tosetti<sup>3</sup>

<sup>1</sup> Web Models

Piazzale Gerbetto, 6. 22100 Como, Italy

{roberto.acerbis, aldo.bongio}@webratio.com

<sup>2</sup> Politecnico di Milano

Piazza L. Da Vinci, 32. 20133 Milano, Italy

{mbrambil, tisi, ceri}@elet.polimi.it,

<sup>3</sup> Acer Europe Services

Via Lepetit, 40. 20020 Lainate (MI), Italy

emanuele\_tosetti@acer-euro.com

**Abstract.** This paper addresses the problem of developing enterprise-class eBusiness solutions in a more economically viable and time-effective way, by adopting Model Driven Development (MDD). Specifically, we report on an experience of more than six years of collaboration between Acer Inc. (the 4th branded PC vendor worldwide) and Web Models, an Italian startup company spinoff of Politecnico di Milano, innovator in the market of software tools and methodologies for MDD. The results clearly demonstrate that MDD can shorten the development of complex eBusiness solutions, improve the quality and conformance to requirements, and increase the economic profitability of solutions, by lowering the total cost of ownership and extending the life span of systems.

**Keywords:** Model Driven Development, WebML, industrial case study.

## 1 Introduction

The advent of the Web as a universal platform has initially facilitated the shift towards enterprise integrated applications, by offering a standard means of distributing data and functions. However, the unprecedented speed of evolution typical of the Web and the fierce competition on technologies among the major ICT players challenges the long-term sustainability of IT projects, due to a number of cross-cutting complexity factors: the spectrum of relevant standards and architectures constantly increases; the learning curve of technologies is higher than their evolution rate; outsourced or distributed development challenges classical software project management.

A solution to the growing complexity of IT projects requires a thorough innovation of the approach to software development, as advocated by the modern research on software engineering, which proposes Model Driven Development (MDD) as a means of improving the governance and end-to-end productivity of software [10]. In essence, MDD promotes a novel approach to software development centered around the

notions of: *platform independent model (PIM)*, which is a representation of the system's functionality independent of any specific technology, and *model transformation*, which is the process of progressively refining high-level models into lower-level ones, until an executable model on a concrete platform is reached.

This paper elaborates on a six-years experience of applying MDD to a set of enterprise-scale applications, developed by the EMEA branch of Acer Inc. using WebRatio, an innovative MDD methodology and tool suite based on the WebML meta-model[2]. The milestones of the work can be summarized as follows:

1. 2000-2002: The introduction of MDD methods and tools in the company, as a means of mastering the deep internal reorganization and technology change.
2. 2002-2004: The consolidation of the MDD approach as a key success factor in mission-critical applications, which led to its diffusion outside the marketing, e.g., to the distribution channel management, sales, and financial services sectors.
3. 2004-today: The integration of the MDD approach within the design of a global Service Oriented Architecture, with the aim of managing large-scale projects, involving not only internal roles, but also distributors and partners.

Model Driven Development seemed the most adequate methodology for bringing software development to non-IT business units. The idea was to exploit the knowledge about the business processes of the marketing personnel, delegating as much as possible of the construction of the implementation code to suitable development tools.

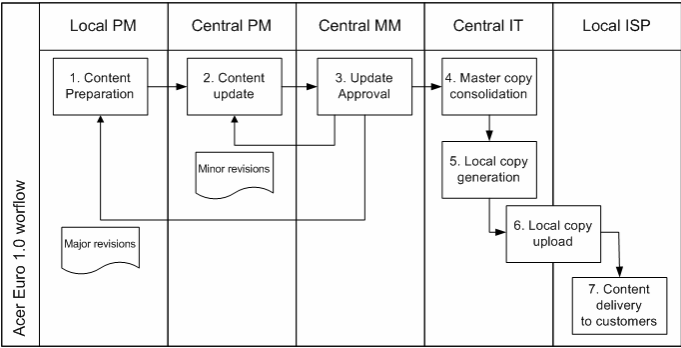
The reference model chosen by Acer is WebML, a model-driven methodology that builds on several previous proposals for hypermedia and Web design notations, including HDM, HDM-lite, RMM, OOHDM, and Araneus[1]. The design principles, notations, and development procedures of WebML are described at large in [2]; HDM [6] pioneered the model-driven design of hypermedia applications and influenced several subsequent proposals like RMM [7], Strudel [4], and OOHDM [8]; while other approaches (e.g., [3]) were inspired by object oriented models. All these methods offer powerful built-in navigation constructs, as opposed to WebML, which exploits simple, yet orthogonal, composition and navigation primitives.

## 2 Case Studies

**Acer-Euro.** The first version of the Acer-Euro application (Acer-Euro 1.0) aimed at establishing a software infrastructure for managing and Web-deploying the marketing and communication contents of 14 countries out of the 31 European Acer national subsidiaries. The Acer-Euro 1.0 system supported the two main functions of Content Publishing and Content Management in a seven-steps distributed workflow, illustrated in Figure 1, involving Local and Central Product Managers (LPMs and CPMs), Central Marketing Managers (CMMs), the central IT department, and the Internet Service Providers (ISPs). In this way, Acer could completely renovate the content and workflow of the marketing and communication functions, while reusing the existing national Internet infrastructures and contracts.

Acer-Euro 1.0 had a very tight schedule. Only seven weeks elapsed from the approval of the new site map and visual identity to the publishing of the 14 national web sites (plus the CMS). In this period, two distinct prototypes were formally approved

by the management: prototype 1, with 50% of functionality (end of week 2), and prototype 2, with 90% of functionality (week 5). Overall, 9 prototypes (2 formal, 7 for internal assessment) were developed in 6 weeks. The development team consisted of four persons: one business expert and one junior developer from Acer, and one analyst and one Java developer from Politecnico di Milano.



**Fig. 1.** Acer-Euro 1.0 workflow

As Table 1 clearly shows, the most relevant aspect of the development of Acer-Euro 1.0 is the short time-to-market with respect to the complexity of the application. Such result has to be ascribed to: *(i)* the high degree of automation of the process thanks to the model-driven approach (90% of the code were synthesized automatically); *(ii)* the overall productivity rate of 131,5 *function points/staff month*, which is 30% greater than the maximum value expected for traditional programming languages in the SPR Tables [9]; *(iii)* the velocity in focusing the requirements, thanks to the rapid production of realistic prototypes; *(iv)* immediate stress test and architecture tuning thanks to code directly generated for the actual delivery platform. Moreover, the benefits of MDD were even more sensible in the maintenance and evolution phase, leading to four major releases between 2001 and 2003 and to 13 multi-language intranet and internet applications, serving additional corporate processes.

**Acer Connect.** In June 2001, a spin-off project, called Acer Connect was scheduled, to address the delivery and management to the sales channel operators (Acer partners). This is a multi-actor extranet application characterized by: segmentation of the users into a hierarchy of user roles; different access privileges and information visibility to roles; one centralized and several local administration roles, able to perform advanced administrative and monitoring tasks; several group-tailored Web applications (e.g., sales, marketing) targeting contents to corporate-specific or partner-specific communities; a security model for managing group and individual access rights on single pieces of contents; full-fledged content personalization.

The first version of Acer Connect was deployed in Italy and UK in December 2001, after only seven elapsed months of development and with an effort of 24 man months. Today, Acer Connect is rolled out in 25 countries, delivering content and services to a community of over 80.000 users. Acer Connect and Acer-Euro share part of the marketing and communication content, and therefore the former project was

realized as an evolution of the latter. The model-driven approach greatly reduced the complexity of integration, because the high-level models of the two systems were an effective tool for reasoning about the functionality to reuse and develop.

Besides Acer-Euro and Acer Connect, several other projects were spun-off, to exploit the customer and partner communities gathered around these two portals, which collectively totalize over 10.800.000 visits per month. As a remark on the long-term sustainability of MDD, we note that, despite their complexity and multi-national reach, both Acer-Euro and Acer Connect are maintained and evolved by one junior developer each, working on the project at part time.

**Table 1.** Main dimensional and economic parameters of the Acer-Euro project

Class	Dimension	Value
Time & effort	Number of elapsed workdays	49
	Number of development staff-months (analysts and developers)	6 staff-months
	Total number of prototypes	9
	Average elapsed man days between consecutive prototypes	5,4
	Average number of development man days per prototype	15,5
Size	Number of localized web applications	14 B2C, 4 CMS
	Number of supported languages	12
	Number of data entry masks	39
	Number of automatically generated database tables	46
	Number of automatically generated database views	82
	Number of automatically generated database statements	279 queries, 89 updates
	Number of automatically generated JSP page templates	48
	Number of automatically generated or reused Java classes	250
	Number of automatically generated Java lines of code	12500
	Number of manually written SQL statements	17 constraints
Degree of automation	Percentage of automatically generated SQL code	96%
	Number of manually written/adapted Java & JSP components	10% JSP
	Percentage of automatically generated Java and JSP code	90% JSP, 100% Java
	Total cost of software development of first version	75.000 €
Cost, ROI, and productivity	HW, SW licenses, and connectivity cost of first version	70.000 € (db)
	Return on investment of first version	12-15 months
	Average effort of extension to one additional country	0,5 staff-months
	Average cost of extension to one additional country	7.500 €
	Average ROI of extension to one additional country	2 months
	Number of function points	789
	Average number of function points delivered per staff-month	131,5

### 3 Results and Critical Considerations

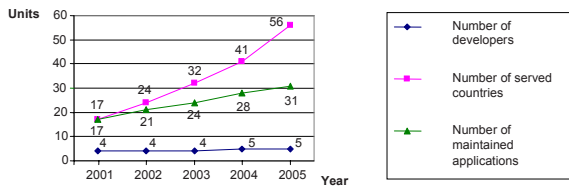
In this section, we summarize the main lessons learned in the application of the MDD principles to web development. The major effect of MDD is to shift the focus of development from implementation to requirement analysis. Almost 80% of the delivery effort concentrates in the phases of data design, hypertext design and prototyping. This means that more development time is spent with the application stakeholders, to refine design models and evaluate prototypes. The result is a better quality of the delivered applications and a higher rate of acceptance, because design errors and requirements misinterpretations are eliminated as early as possible. MDD also benefits the more technical tasks of testing, maintenance, and evolution, because reasoning on the system is far more effective at the conceptual level than at the physical level.

MDD lowers the technical barriers for developing complex Web applications, allowing a more flexible distribution of responsibilities between the IT department and the business units. When business goals are rapidly evolving, and quick adaptation to changing environment is a critical factor, the possibility of developing, monitoring and adjusting the systems directly by the business units greatly improves efficiency.

The deployment consisted of J2EE standard architectures, with the integration of heterogeneous systems taking place by means of Web services. A well-defined architectural protocol can be established to integrate systems autonomously developed in different business units, avoiding the duplication of software functions and data.

Last but not least, MDD has proven an economically profitable and sustainable way of developing Web systems. The peak productivity rates experienced in the Acer projects has reached five times the number of delivered function points per staff-month of a traditional programming language like Java [9].

On the negative side of MDD, the initial training costs must be considered. MDD requires non-technical knowledge on the modeling of software solutions, which must be acquired with a mix of conventional and on-the-job training. Acer estimates that it took from 4 to 6 months to have fully productive developers with MDD, WebML, and WebRatio. However, as Figure 2 shows, the initial investment in human capital required by MDD pays off in the mid term. The number of applications developed and maintained per unit of personnel increases with the developers' expertise and exceeds ten fully operational, complex and distributed Web applications per developer.



**Fig. 2.** Evolution of manpower versus number of maintained applications and served countries

## 4 Conclusions

In this paper we have reported on a long-term experience in applying Model Driven Development to the construction of mission-critical eBusiness solutions, jointly conducted by Acer and WebModels, by exploiting the WebML model. After more than five years of applying MDD with WebRatio, Acer has gained sufficient experience to draw some general conclusions. Today, the use of WebRatio has spread from the Acer-Euro project to most of the Web-based B2C, B2E, and B2B platforms of Acer EMEA and has been exported from Europe also to Acer PanAmerica. The developed solutions cover all the most critical sectors of Acer's business and have given tangible benefits over the years. The abovementioned portfolio of solutions has been deployed, and is continuously being maintained, by an internal team consisting of five developers only. With a traditional development methodology and using conventional programming-oriented tools, the company estimates that the construction of the deployed systems would have required at least three times the resources that have been

invested. None of the developed systems has been retired or has become obsolete; new requirements have been smoothly incorporated into the running versions and rolled out by iteratively extending the deployed systems.

The Acer experience has demonstrated the feasibility of MDD and the efficiency it introduces into the development lifecycle, largely anticipating the current debate on the Model Driven Architecture. However, the adoption of a model-based approach must also extend to the maintenance and evolution steps (which account for over 60% of the total lifecycle cost), where they provide the best advantages in terms of cost and productivity.

In conclusion, MDD appears to be a powerful tool for renovating businesses and taking advantage of the advent of low-cost distributed network infrastructures. However, the transition requires innovation not only in the business strategies but also in the IT departments.

Our future work will concentrate on improving and extending the quantitative assessment of the benefits of Model Driven Development in the Web application sector. A novel software tool for automatically performing the evaluation of software projects size and effort is under development, which will support the measurement of different project parameters related to size, effort, and cost.

## References

- [1] Atzeni, P., Mecca, G., Merialdo, P.: Design and maintenance of data-intensive Web sites. In: Schek, H.-J., Saltor, F., Ramos, I., Alonso, G. (eds.) EDBT 1998. LNCS, vol. 1377, pp. 436–450. Springer, Heidelberg (1998)
- [2] Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M.: Designing Data-Intensive Web Applications. Morgan Kaufmann, Seattle, Washington (2003)
- [3] Conallen, J.: Modeling Web Application Architectures with UML. *Communications of the ACM* 42(10), 63–70 (1999)
- [4] Fernandez, M., Florescu, D., et al.: Catching the boat with Strudel: Experiences with a Web-site management system. In: 24th ACM SIGMOD, Seattle, WA (1998)
- [5] Fraternali, P.: Tools and approaches for developing data-intensive Web applications: A Survey. *ACM Computing Survey* 31(3), 227–263 (1999)
- [6] Garzotto, F., Paolini, P., Schwabe, D.: HDM - A Model-Based Approach to Hypertext Application Design. *ACM TOIS* 11(1), 1–26 (1993)
- [7] Isakowitz, T., Stohr, E., Balasubramanian, P.: RMM: A Methodology for Structured Hypermedia Design. *CACM* 38(8), 34–44 (1995)
- [8] Rossi, G., Schwabe, D., Lyardet, F.: Web Application Models are More than Conceptual Models. In: Kouloumdjian, J., Roddick, J.F., Chen, P.P., Embley, D.W., Liddle, S.W. (eds.) *Advances in Conceptual Modeling*. LNCS, vol. 1727, pp. 239–252. Springer, Heidelberg (1999)
- [9] Software Productivity Research: SPR Programming language Table – Version PLT2005a (2005) <http://www.spr.com>
- [10] Warner, J., Bast, W., Pinkley, D., Herrera, M., Kleppe, A.: MDA Explained: The Model Driven Architecture: Practice and Promise. Addison Wesley, London, UK (2003)