Chapter 8. Web Engineering

Table of Contents

MODEL-DRIVEN WEB ENGINEERING - MAGIC OR REALITY?

Modern Web applications are full-fledged, complex software systems. Due to the evolution of Web technologies the Web has become a primary platform for developing applications. However, as these technologies evolve very fast, they might become obsolete soon. Developers of Web applications need sophisticated solutions that support the whole product lifetime of an application that is able to cope with the quick changes of the underlying technologies.

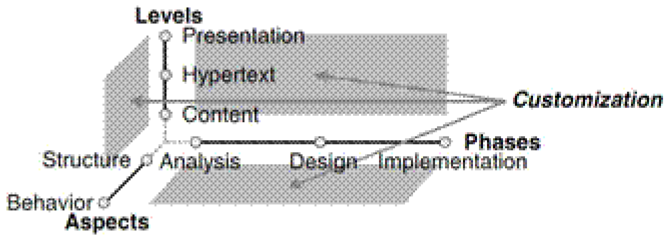
Therefore, the development of Web applications requires a methodologically sound engineering approach. Model-driven Web Engineering (MDWE) is a still emerging field aiming at providing sound model-based solutions for building Web applications that try to separate the abstract design (PIM) from the concrete technological platforms (PSMs).

Introduction

The evolution of Web technologies increased the presence of the Web in our everyday life starting from personal home pages through corporate portals and Web shops up to Web applications implementing complex business processes. This diversity of applications makes the selection of the appropriate technology and platform harder, in particular, when these applications should collaborate with each other. Some of them might be suitable in a given case, which are sometimes alternatives to each other, while they should be combined in other cases. As these technologies evolve very fast, they might become obsolete soon. It is very hard to predict, which technologies will be out tomorrow so business logic should be as independent as possible from technologies to allow change in the technology without affecting business processes and rules.

The modeling of such systems is a very complex process since (in an ideal case) it should take both existing and future technologies into consideration in order to create a flexible system which allows further development. Despite changing of the technology, models of the application domain remain almost the same since they capture business model and processes. The best practice tells us to keep the modeling of the application domain and the technological details separated.

Existing model-based Web Engineering approaches provide different methods and tools for both the design and the development of various kinds of Web applications. In order to reduce complexity, most of the methodologies propose the separation of different views (i.e., models) of the application into 3 levels: structural (or content), navigational (or hypertext) and presentational models. For more information see [7]. Figure 1 shows the most common design dimensions of the currently existing methodologies.



Design dimensions of Web applications.

In addition, some methodologies add some new models (or refine existing ones) to obtain a more fine-grained solution when modeling the application. Despite the separation, the levels should be interconnected in order to be able to capture the semantics behind the elements of the different models, e.g., the navigational objects are based on certain elements of the content model.

Beyond the creation of the models for the corresponding levels, Web application designers need to be aware of the various aspects of the systems to be modeled. Some applications are providing access to more or less static information hence they require much less behaviour modeling compared to systems that need to perform several complex business processes like e-commerce applications. Both structure and behaviour need to be modeled using a uniform notation that has to cope with the specific characteristic of each of the levels.

Current design methods offer some possibilities for modeling the levels and aspects mentioned above but they all has a unique approach (e.g., offering some model kinds that the others not) so this field is not standardized.

There is an other approach worth mentioning when talking about Web application design. Let this be either a fortune or an unfortune, there is no consensus in the literature about the general phases of the development which means that the order of steps involved in modeling the levels is up to the modeler.

MDA & MDE

The Model-Driven Architecture (MDA) is a software design approach that was officially launched in 2001. MDA is intended to support model-driven engineering of software systems. The MDA is a specification that provides a set of guidelines for structuring specifications expressed as models. Using the MDA methodology, system functionality may first be defined as a platform-independent model (PIM) through an appropriate Domain Specific Language. Given a Platform Definition Model (PDM) corresponding to CORBA, .Net, the Web, etc., the PIM may then be translated to one or more platform-specific models (PSMs) for the actual implementation, using different Domain Specific Languages, or a General Purpose Language like Java, C#, Python, etc. The principles of MDA can also be applied to other areas like business process modeling where the architecture and technology neutral PIM is mapped onto either system or manual processes.

One of the main aims of the MDA is to separate design from architecture and realization technologies facilitating that design and architecture can alter independently. The design addresses the functional (use case) requirements while architecture provides the infrastructure through which non-functional requirements like scalability, reliability and performance are realized. MDA envisages that the platform independent model (PIM), which represents a conceptual design realizing the functional requirements, will survive changes in realization technologies and software architectures. In other words, ”Design once, build it on any platform”.

MDA at a glance.

Following a long history of the use of models to represent key ideas in both problem and solution domains, MDA provides a conceptual framework for using models and applying transformations between them as part of a controlled, efficient software development process. The basic assumptions that governing MDA usage:

- Models help people understand and communicate complex ideas.
- Many different kinds of elements can be modeled, depending on the context. These offer different views of the world that must ultimately be reconciled.

- We see commonality at all levels of these models in both the problems being analyzed and the proposed solutions.
- Applying the ideas of different kinds of models and transforming them between representations provides a well-defined style of development, enabling the identification and reuse of common approaches.
- Provides a conceptual framework and a set of standards to express models, model relationships, and model-to-model transformations.
- Tools and technologies can help to realize this approach, and make it practical and efficient to apply.

Model Driven Engineering.

In order to raise the level of abstraction in program specification and increase automation in program development we need to use engineering approaches as well. The idea promoted by Model Driven Engineering (MDE) is to use models at different levels of abstraction for developing systems, thereby raising the level of abstraction in program specification. An increase of automation in program development is reached by using executable model transformations. Higher-level models are transformed into lower level models until the model can be made executable using either code generation or model interpretation.

MDE is often confused with Model Driven Architecture. MDA can be seen as OMG's (Object Management Group) vision on MDE but MDE is wider in scope than MDA. MDE combines process and analysis with architecture. The MDE approach is meant to increase productivity by maximizing compatibility between systems (via reuse of standardized models), simplifying the process of design (via models of recurring design patterns in the application domain), and promoting communication between individuals and teams working on the system (via a standardization of the terminology and the best practices used in the application domain). A modeling paradigm for MDE is considered effective, if its models make sense from the point of view of a user that is familiar with the domain, and if they can serve as a basis for implementing systems. The models are developed through extensive communication among product managers, designers, developers and users of the application domain. As the models approach completion, they enable the development of software and systems.

Prev Up Next
Part III. Web Engineering Home Domain Specific Models and Languages