

Malgorzata Plechawska-Wójcik¹

A MODEL-DRIVEN APPROACH TO CONSTRUCTION OF WEB APPLICATIONS BASED ON BPMN AND WEBML SOLUTIONS

The article presents the use of modern model driven engineering (MDE) technologies in development of web-based solution. Integration of BPMN and WebML constitutes a fast and efficient way to create web applications. The BPMN application model can be automatically transformed to WebML model. The data model also needs to be created.

Keywords: WebML, BPMN, MDE, model-driven.

Малгожата Плехавська-Вуйчик

МОДЕЛЬНО-ОРІЄНТОВАНИЙ ПІДХІД ДО СТВОРЕННЯ ВЕБ-ДОДАТКІВ НА БАЗІ РІШЕНЬ BPMN І WEBML

У статті представлено методи використання сучасних технологій модельно-орієнтованої розробки (MDE) у розвитку веб-рішень. Інтеграція BPMN і WebML є швидким і ефективним способом створення веб-додатків. Модель додатка BPMN може бути автоматично перетворена в модель WebML, також має бути створена модель даних.

Ключові слова: WebML; BPMN; MDE; модельно-орієнтований.

Малгожата Плехавська-Вуйчик

МОДЕЛЬНО-ОРИЕНТИРОВАННЫЙ ПОДХОД К СОЗДАНИЮ ВЕБ-ПРИЛОЖЕНИЙ, ОСНОВАННЫХ НА РЕШЕНИЯХ BPMN И WEBML

В статье представлены методы использования современных технологий модельно-ориентированной разработки (MDE) в развитии веб-решений. Интеграция BPMN и WebML представляет собой быстрый и эффективный способ создания веб-приложений. Модель приложения BPMN может быть автоматически преобразована в модель WebML, также должна быть создана модель данных.

Ключевые слова: WebML; BPMN; MDE; модельно-ориентированный.

1. Introduction. Model-driven engineering (MDE) [2] is an innovative approach to software development. It is based on the models, which are used to develop software artifacts, including code, documentation and tests. The model-driven architecture (MDA) strategy was developed by the Object Management Groups (OMG). MDA concept assumes that models plays the main role in software development [5]. MDE, however, is a wider idea than MDA. It covers not only the process of development but also the architecture analysis. The idea of MDA assumes separation of the specification of system functionalities from its implementation on a specific technology platform. MDA is focused on models, artifacts and architecture.

Using models in software development has many advantages. It helps to reduce costs and to improve quality. Models can be easily manipulated and transformed. They also can be an appropriate tool for solving integration problems, because of interoperability specifications [5]. The same model of system functionalities can be

¹ Institute of Computer Science, Lublin University of Technology, Poland.

realized on multiple platforms. It is useful especially in case of integration or migration of a system to different platform technologies. OMG distinguishes two models: PIM (platform independent models) and PSM (platform specific models). Platform-independent models are easier to validate and test. It is also easier to develop implementations on different platforms having defined one precise model of the system. Platform-independent models should have a proper level of abstraction [4]. Usually they are also more clear and easy to use. Such independent models are used to summarize system functionalities and give alternative perspectives of the system. They can be also easy mapped down to platform specific mechanisms [5]. Formal conceptual system models solve problems of interoperability, integration and management of metadata across applications, platforms and databases [1].

The key idea of MDE is to use model transformations. OMG assumes that model transformations is a tool for developers. The MDA approach distinguishes 3 elements used in model transformation [1,12]:

- meta-meta-model - the basis of a model providing the language used for the meta-model description;
- meta-models of the models to transform - source and target models correspond to meta-model; modeling is done according to the meta-meta-model;
- source and target models - specific instances of the corresponding meta-models.

Model transformations need to have defined source and target models regarding the meta-meta-model and a map source. Formal techniques of transformations should also be known.

2. BPMN models. Business process modeling notation (BPMN) is a standard developed by the Business Process Management Initiative (BPMI) in 2004 [10]. The idea of BPMN was to create a notation clear to business users, such as business analysts, technical developers, programmers and managers. BPMN can be used to model drafts of processes. It constitutes a bridge between business process design and process implementation [10].

Models created in BPMN are flowcharts of graphical models describing business process operations. This network of graphical objects contains activities describing particular processes and flow controls defining the order of performance.

Graphical elements which are defined in the BPMN notation have a similar shape of those elements, which can be found in typical business models. For example, among typical elements one can find activities marked as rectangles and decisions - diamonds. BPMN elements look familiar to business analysts and that is why they should be easily used and adapted.

The idea of BPMN was to create a simple, easy to use notation, which can be used to handle even very complex business process models. To make it easier to use the notation, its elements are organized into several categories which help to recognize the elements and place them correctly at the diagram. Those categories are:

- Flow objects - core BPMN elements which are: activities (model tasks and processes), events (indicate on triggers or results) and gateways (used for the flow control, represent decisions, forking and merging of paths).
- Connecting objects - linking lines creating the basic structure of the processes. There are sequence flows (show the order of activities), message flows (represent

flow of messages between different process elements) and associations (linking notes, texts and data with flow objects).

- Swimlanes - a mechanism used for organization of the model activities into separated categories. There are pools (representing a single participant in the process) and lanes (sub-partitions of a pool, used to categorize activities).

- Artifacts - additional elements complementing and extending the model. The most popular are: data objects (show the data which are collected and required by activities), groups (support the documentation and analysis) and annotations (represent comments).

The example of BPMN process adapted to WebML model is presented in details at [8].

3. WebML and WebRatio. WebRatio is a tool which can be used for designing models in BPMN and WebML (the Web Modeling Language). WebRatio represents an approach of system development based on models designing. The whole application is based on designed models. There is no necessity to write a code - everything is based on models and its properties.

All WebRatio projects are based on two types of models: the process model and the application model. One can use one of them or both, depending on the application type.

The process model is based on the BPMN. It is useful in defining users, business roles, activities and tasks assigned to actors or business roles. It is also possible to define constraints and conditions of the workflow.

The application model is based on the WebML. This language enables to define [3]: data, business and page composition logic, navigation and interaction concepts, integration with external services and components.

WebML is a formal, graphical, high-level description language which can be used to build and maintain the complete project. System development is supported by a visual design tool. It is possible to manipulate data, model users, compose the content into pages, model navigation, store and maintain meta-data, personalize policies and permissions. Thus, WebML models contain the hypertext model, the structure of the page, its composition, navigation as well as content management specification and external services integration mechanisms.

WebRatio introduces 3 types of modes: data model, hypertext model and presentation model.

3.1. Data models. WebML data model is a typical, commonly known ERD (entity-relationship diagram) model. It is based on entities and relationships defined between them. It is possible to display an UML view or WebML view, which is inverted (Fig. 2).

Each entity is given a unique identifier (OID). WebML OIDs [3] are abstract concepts and can be implemented in relational databases or in XML data.

WebML data model use Hibernate framework. It accepts generalization, it is possible to import tables from other databases. The data model enables also modeling structures for those types of databases which are supported by the Hibernate framework. It is possible to use sequences and contain images. A developer can add derived attributes and relationships. In practice, they are realized with appropriate views. It is also possible to use N:N relationships, which are automatically transformed into 1:N and N:1 relationships (a transition table is created).

WebRatio uses user-group-module pattern [6] to handle personalization. User is assigned to a group, which has its own default module. After the user is logged to the application, the proper view is displayed. This personalization scheme sets proper behavior of the application and it is the only way to perform it properly. Entities names (users, groups, modules) are predefined and there is no possibility to avoid using them. Those entities names (especially users) are restricted in some databases and it can cause problems (fortunately, it is possible to change their names) [11]. Users and groups managing, especially an admin area is usually done manually. However, it is also possible to use a wizard. Several detailed examples of the data models can be found at [9].

3.2. Hypertext models. The hypertext model of the site is responsible for the logic. It covers composition and navigation and describes content of the project pages. The model can contain different user views, each has several pages. The single page is a container storing a different kind of elements such as data or entry units, links, navigations and operations. The entire model can contain [7]:

- Units as atomic elements used to publish information, display forms and interact with a user. There are several groups of units. Each data unit can be associated with an entity or relationship and it enables handling its content. Units can be used to display data of a single object (single data units) or a set of them (multidata, index, scroller or hierarchical units). There are also data entry units representing forms containing a set of fields with such features as input validation, fields preloading and auto-completion based on AJAX. Selectors attached to data units are used to specify additional data condition and restrictions on attributes values.

- Operations are actions triggered by a user (for example, adding or modifying data, logging, connecting, sending emails). They can have input and output operations, but they do not display any information and they are placed outside a page. It is also possible to create own operations based on groovy scripts.

- Links are used to organize navigation between units inside single page as well as between pages. Links can carry information (contextual links) or can be only navigational (non-contextual) between pages. Links are the only way of data exchange between units. It is possible to use OK and KO links, representing respectively successful and failed operations. There are also automatic links (used if a unit needs to be displayed as soon as the page is accessed) and transportable links (used only to pass data, not to navigate).

- Areas and site views are used to organize the application hierarchy and introduce better granularity. Areas are used as a sections grouping pages according, for example, to business purposes. Site views are dedicated to a specific type of users (for example, administrators, users or customers) and allows setting authentication requirements.

Pages and areas can have a landmark property [6] useful in navigation concept creation. This property is responsible for direct displaying the object in the automatically generated menu, which can be hierarchical and contains landmarked areas and pages of the particular site view.

3.3. Presentation models. A presentation model is responsible for defining the site views look. WebML does not provide an easy-to-use tool for presentation layer. It

needs to be developed by attaching style sheets to pages, site views and units. Style sheets should have XSL format. It is also possible to attach predefined presentation style sheets and the server-side components.

It is possible to define layout for each unit, page and entire model. Single project can contain several layouts [11]. The default layout set to a model or a page can be easily overridden by each model element. A user can also use groovy scripts.

4. Application development. WebRatio supports development of application with automatic form creation. Wizards create fields and connect them to chosen entities. Links and operations, however, must be added manually.

Personalization of the application is easy to perform using the hypertext model and a user-group-module pattern. The menu generated automatically based on landmarks gives access to areas to which the user has the access rights and hides links to the forbidden areas. Access to protected pages is protected by a password. It is also possible to display different pages depending on a performed operation and its result.

WebRatio, which extends the WebML, gives possibility to develop web application and see results only by changing the model, without programming. However, the complete view is complicated and it is composed of many elements as areas, pages and units connected with links sharing parameters.

The fragment of the hypertext, data and BPMN model is shown in figures available on the Internet: pluton.pol.lublin.pl/~gosiap/materialy/MPlechawska_figs.rar. The presented application is dedicated to Internet registration of the patients to a medical clinic. The BPMN model illustrated in Fig. 1 presents a process of the registration. Figures present a part of the WebML model automatically generated on the basis on BPMN. Also the data model was generated automatically. It is possible to add changes to both data and hypertext models. The database needs to be configured manually. Also database connection and data need to be added by hand. The presentation model also needs to be configured. However, it is possible to adapt predefined models.

5. Summary. WebRatio is a tool supporting a model-driven approach. This approach enables rapid web application development based on WebML and BPMN. However, traditional programming methods of software development give more freedom in developing and managing the contents. WebRatio is useful especially in cases of simple application based on CRUD forms. Development of more complicated systems is also possible. However, it needs more sophisticated, composed models dealing with many units and parameters.

References:

- Almendros-Jimenez, J. M., Iribarne, L.* (2009). UML Modeling of User and Database Interaction. *The Computer Journal*, 52(3):348-367, 2009.
- Balasubramanian, K., Schmidt, D., Molnar, Z., Ledeczki, A.* (2009). *System Integration Using Model-Driven Engineering, Designing Software-Intensive Systems: Methods and Principles.*
- Brambilla, M., Butti, S., Fraternali, P.* (2010). WebRatio BPM: A Tool for Designing and Deploying Business Processes on the Web, *WEB ENGINEERING, Lecture Notes in Computer Science*, 2010, Volume 6189/2010, 415-429, 2010.
- Gomez-Lopez, M., Reina-Quintero, A., Gasca, R.* (2011) *Model-Driven Engineering for Constraint Database Query Evaluation, MELO 2011 - Model-Driven Engineering.*
- Kent, S.* (2002). *Model-driven engineering. Integrated formal methods. Lecture Notes in Computer Science*, 2002, Volume 2335: 286-298.
- Kesik, J., Zyla, K.* (2011). Usability comparison of WebRatio and Symphony for educational purposes, *Proceedings of Electrotechnical Institute, Issue 247*, 2011.

- The Web Modeling Language <http://www.webml.org>
- Web Ratio: The Tailor Of BPM Applications: http://www.webratio.com/Portal/casestudies_pdf/WebRatio_TheTailorOfBPMApplications.pdf
- WebML.org - The Web Modeling Language resources: web models and hypertext models, <http://webml.org/webml/page33.do?UserCtxParam=0&GroupCtxParam=0&ctx1=EN>
- White, S. A.* (2009) Business Process Modeling Notation (BPMN) Version 1.0. Business Process Management Initiative BPMIorg May. Retrieved from <http://www.omg.org/spec/BPMN/1.2>.
- Zyla, K., Kesik, J.* (2011) BPMN and WebML solution for web app-aided flow of documents, New materials and IT technologies in production engineering, 114-125.
- Zyla, K.* (2012). Wykorzystanie mechanizmow lokalizacji urzadzenia mobilnego w oparciu o Google App Inventor. Logistyka 3/2012, Instytut Logistyki i Magazynowania, Poznan.

Стаття надійшла до редакції 27.07.12