USABILITY COMPARISON OF WEBRATIO AND SYMFONY FOR EDUCATIONAL PURPOSES

ABSTRACT
The teaching of web design techniques is strongly influenced by the software producers, who want to gain well qualified programmers with appropriate knowledge regarding model driven and traditional techniques of software development.

This paper describes a comparison of two MVC based frameworks chosen for web design course – WebRatio and symfony.

The comparison points out the pros and cons of frameworks, revealed during the conduction of courses. It also gives hints how to solve the problem of choosing appropriate tools for using on different fields of web design techniques teaching, elaborated by the autors during courses conducted for the students who have to acquire abilities demanded by the modern work market.

Słowa kluczowe: web design teaching, WebML, WebRatio, symfony

1. INTRODUCTION

Nowadays application development concentrates on the web. The growing need for the new web based applications causes a demand for the methods
of rapid web-app development and for the professionals knowing how to implement them. Consumers and developers expect the fast realization of their requests while keeping in mind application’s security, scalability, usability, efficiency etc.

While serving many different purposes, those applications are quite similar inside and follow some basic rules of the web-app design. The MVC model of such design has almost become a standard. The Model provides data access, the View takes care of data presentation and user interface and the Controller contains business logic connecting first two ones. A number of frameworks has been developed basing on that. Their common purpose is to speed up and ease up web-app development by providing a set of commonly used operations in a ready to implement state. The inner design of such frameworks forces the programmer to follow the rules of the MVC model.

This paper is a comparison of the two MVC based frameworks, representing different approaches to the web design. It considers the study of usefulness of these frameworks with the teaching of the MVC oriented web development at the higher education level.

Symfony represents a programmer-like approach to the application development. Most of the work involves a coding in the PHP or YML language. On the other hand it is fully customizable, including easy source code edition, so it is suitable for the less-standard tasks.

WebRatio represents a designer-like approach to the application development. Application is generated on the basis of the two models being a connection of the predefined chunks (units). The first one embraces Model part of the MVC while second one contains VC part of it, thus being a M(VC) approach. These models are up-to-date documentation expressing the exact state and structure of the application. On the other hand there is only a small number of the predefined chunks, limiting the freedom of non-standard application development. There exists a possibility to develop the custom units but it causes a return to the programmer approach, in this case JSP.

In the authors eyes, these two frameworks represent the main approaches in the courseware of the web (or other applications) design. While the first one is widely used and known, the second one is relatively new and worth to be compared to the common teaching methods.

2. USING WEBML WITH WEBRATIO

The WebRatio framework introduces a visual modeling language called WebML. It is described by its authors as “a visual notation for specifying the
content, composition, and navigation features of hypertext applications, building on ER and UML” [1]. It can be seen as a high level visual description of a MVC organized web application.

WebML consists of three models:
- Data Model,
- Hypertext Model,
- Presentation Model.

Data model describes application data design and is similar to a standard ERD model used in conceptual database design.

![Fig. 1. The example of a hypertext model](image)

Hypertext model concentrates on a composition and navigation of the web-app. It arranges application into views containing areas and pages. Pages
contain units responsible for the data presentation. The business logic is projected by the operation units placed either within or outside the pages (Fig. 1). The set of units representing the commonly used operations is made available. Non-standard operation can be introduced by defining the custom units. Pages and units are connected by the links representing information flow between them. Links can carry the activation and be rendered on the page, or be in so called “transport” mode, representing the inner data flows inside application. The hypertext model is independent from the data model, however it uses the names of the entities and attributes described by it [1].

The third layer is the presentation model describing the layout of the data presentation units on the pages, the pages layout and the layout of the whole application. It imports the standard DOM/CSS design and is partially left for the developers consideration making it a bit fuzzy, on the other hand leaving the freedom to use the presentation technique of developers choice.

The WebRatio IDE out of the box provides: designer tool, Apache Derby Database, Apache Tomcat (application server). It transfers WebML concepts to the real web application development and enables automatic application generation basing on the WebML model aided by a few own refinements. Resulting application is output in a JSP code and follows Apache Struts MVC pattern.

The WebRatio framework handles the construction of a WebML Data model as well as the connection and creation of the corresponding database via Hibernate. It also allows to import an existing database into the model. The Hypertext and Presentation model is unified into the Site or the Service view. Both of these allow to visually construct a site Hypertext model starting from views, through areas and pages, ending with units, their attributes and connections. The Presentation model is handled by layout definitions of pages, as well as single units. The units organization on the page is handled by the Grid feature, resulting in the old way, table organized web page. It is also possible to use layers layout but it requires to manually setup needed areas [5].

Besides handling WebML notation, the WebRatio supports user identification and access control. It is organized in a strict pattern: User – Group – Module, where User belongs to a group (or groups) which has access to a module (or modules). Module is understood as any part of the hypertext model (including single units) marked as protected. In order to handle distinct Users, Groups and Modules WebRatio forces the creation of the proper set of three entities in the Data Model.

3. USING SYMFONY WITH NETBEANS

Symfony is a complete framework written entirely in PHP5 and designed to optimize and fasten the development of the web applications. It supports
most of available database servers and allows to choose between Propel and Doctrine ORM. Packages containing framework files are downloadable free of charge from the symfony home page [7].

The symfony project can be set up by placing framework files inside the project directory tree, which makes it possible to treat it independently as a whole. After PHP and symfony paths are configured, developer achieves ability to use the set of symfony commands, also called tasks. They have to be written manually in console or invoked by some tool i.e. Netbeans.

Project creation task builds up project directory tree around the framework source files, which allows to make project independent from a version of symfony installed on the web server. Another task creates applications in the project scope (one can create more than one application). Each application has separate directory tree (inside the apps project directory) and two controller files placed in the project web directory. Web directory contains all files that can be visible for client – user of the web application. Almost all configuration files are written using YML language – human readable XML hybrid, providing sets of key/value pairs through the labels and indentations [6].

All requests to application are made through the front controllers, pointing to the same application but different environments. The programmer can create environments for his special purposes or use the default ones [6]:
1. Development environment – used by web developers while working on the application.
2. Test environment – used to automatically test the application.
3. Production environment – used by end users.

Environments have different configuration i.e. in development environment logs and details of requests are shown and the cache is disabled for all changes in the code to be visible right away.

The Netbeans for PHP (PHP developer tool written in Java and requiring Java Development Kit) supports, since version 6.8, the symfony framework. It provides integrated facilities like showing tasks output, automating paths and aliases creation, advanced debugging, syntax prompting and other features, so far available for Java and C++. Netbeans gives the developer an opportunity to use complex prompts regarding symfony commands, therefore there is no need to remember their syntax. After selecting command from the presented list, a manual page with available parameters and examples of usage is displayed. Unfortunately Netbeans IDE is not too efficient i.e. vast number of files (when using SVN number of project files is doubled) slowing down the scanning project for changes, also the earlier versions of IDE are not fully compatible with displaying syntax prompts.
Comparing symfony and WebRatio, the second one allows to quickly start an education process. One, easy installable application is required to develop and test the course exercises. The preparation of the symfony framework environment is complex and quite difficult, however the students can acquire knowledge extending the web development, such as installing and configuring the PHP based Web Server.

4. WORKING WITH DATA LAYER

WebRatio uses ORM Hibernate and JDBC drivers for data management therefore it supports all kinds of data sources provided by JDBC drivers. Drivers for databases other than Apache Derby have to be downloaded from appropriate web pages and put into .\WebRatio5.1.0/WebRatio/Drivers directory. The jar file containing driver classes should have specific name, otherwise it will not be recognized by IDE. Displaying of the expected jar file names for various JDBC drivers would be very useful.

Data model can be projected using easy to understand graphical notation originating from Entity Relationship Diagram and UML class diagram, what simplifies data model perception (Fig. 2). Developer is able to choose between showing relationship cardinality using WebML (ER) style and UML style.

The large advantage of WebRatio is Database Synchronizer which allows to:
1. Synchronize data model with one data source at once.
2. Display the elements of the data model and data source, which are not synchronized.
3. Import the selected entities and relationships into model.
4. Export the selected data model elements into database.
5. Execute generated DDL in transaction.

Symfony is typically used with Propel ORM and MySQL PDO driver. The addition of PDO drivers is quite difficult, when compared to JDBC drivers. On Unix/Linux systems it requires the building of PDO as a shared extension or installing it as a shared module (php.ini file needs to be manually updated). For Windows, PDO and all the major drivers are delivered with PHP as shared extensions, and need to be activated by editing the php.ini file [3].

Configuration of data sources involves running symfony configure:database task or editing databases.yml files. The basis of data model are schema.yml textual files (Fig. 3), which are manually written or built basing on existing data sources structure (propel:build-schema task). Because of lack of graphical representation it is hard to follow the dependencies between entities. All modifications and dependencies actualizations have to be made manually, so large schemas are difficult to manage.

Propel:
SM_Marka: #
id:
nazwa: { type: varchar(100), required: true }importer: { type: varchar(255) }

SM_Marka_model: #
id:
sm_marka_id: { type: integer, foreignTable: SM_Marka,foreignReference: id, required: true }
nazwa: { type: varchar(100), required: true }
rok_modelowy: { type: date, required: true }
ocena_NCAP: { type: tinyint }
koniec_sprzedazy: { type: boolean, default: 0 }

Fig. 3. Representation of the SmMarka and SmMarkaModel entities in the YML language

One of the good security practices is to make database structure hard to understand in case of unauthorized access. When it comes to the symfony, it is not the case. symfony allows developers not to specify properties for some attributes:
• created_at, updated_at, created_on, updated_on – automatically set as the timestamp type,
• id – automatically set as the primary key,
• referencedTable_id – automatically set as the foreign key to the referencedTable.

This feature has great influence on making database structure predictable and exposing the type of the framework which was used to build web application [7]. It can be an example of bad practice shown to students and properly explained for its disadvantages.

Database tables population can be made using fixtures, which are files with initial data (propel: data-load command). Fixtures can be built from the database content, unfortunately information about instances identifiers is lost. Beside that, loading of a large fixtures (about 50k rows) can cause web server to produce out of memory error.

The main advantage of WebRatio is the visual notation of data schematics, which is suitable for quickly grasping the database construction ideas. It eases up the learning process by providing database-synchronized diagrams and wizards. Although differences in building data model can be the basis for showing students the examples of good and bad practices.

5. CONTENT MANAGEMENT

The proper navigation between pages and areas of web application is always a big issue. WebRatio introduces a concept of landmark property ascribed to the pages and presentation units. By turning this property on/off, designer can decide whether the object is directly accessible from the web and represented in the automatically generated Menu. The menu follows the hierarchy of the areas and pages of the particular view of the application.

One of the most common feature of the web application is the form. Forms are used always when gathering the information from the user and each one needs significant amount of labor. WebRatio offers some support with the form creation. First of all the entry unit (WebML representation of the form) is constructed visually as well as the fields inside it. The framework offers a standard set of fields with some key features preprogrammed and ready to activate. These features include: input validation-based on the field type, fields preloading, filling of the lists, performing predefined AJAX actions like field auto completion [2]. On the other hand utilization of a non-standard field behavior requires the knowledge far beyond simple JSP programming.
When it comes to the automation of the form creation, WebRatio provides a wizard, creating fields based on chosen entity. Wizard recognizes the entity relations and creates proper list fields. Unfortunately there is no support for the creation of common tasks like adding or editing entity using the data from the form. All the operations have to be done manually i.e. placing and connecting proper units.

When it comes to symfony, all content management is based on the data model described in `schema.yml` files and established data sources. Dedicated tasks create appropriate classes and configuration files. Depending on actual needs developer can use php pages generators which are optimized for CRUD web applications. Specialized tasks create customizable modules, which contain all classes, forms and configuration files necessary to perform CRUD operations via web browser. These operations include browsing a list of entity instances, instances addition, modification and deletion. When entity contains foreign keys, there are drop-down lists generated, unfortunately it requires manual modifications of `_toString()` method. Developer has to point out which information from related entity should be printed in the list.

Symfony contains some standard validators that can be used for forms [7]. Any other validators have to be defined by a developer as the proper PHP classes with specific methods. Also redisplaying form with error messages and filled in fields requires code modifications. Customized, as well as standard, validators can be used in yml configuration file instead of `validateSend()` method.

That part shows clearly the difference between developer and programmer approach to the support of framework for its user. A model driven (developer like) approach used in WebRatio, gives the feature of rapid development, while (programmer like) approach of symfony gives more freedom in managing the content. It is worth to mention that symfony developers can achieve ready to use CRUD forms (with default layout and actions) faster then WebRatio ones.

6. PERSONALIZATION

WebRatio provides its own way of the site personalization basing on the User-Group-Module pattern (Fig. 4). After successful login the default group of the user is found and he/she is moved to the module, set as the default for that group. Personalization subschema is mandatory for all applications, so there is no way to avoid that behavior and also no way to authenticate the user basing on some other entities. Unfortunately entities names, like user and
group, are restricted in some databases, therefore their correct mapping is undoable (fortunately WebRatio allows to change their names) and their often usage may lead students to confusing errors.

Filling of the personalization entities is a multistep process involving admin area (special tool for inserting into database or deleting information about protected modules) and hypertext model, for managing of the users and groups creation. That part of the model has to be done manually, it could also be implemented as a wizard.

Although it is the one and only way of handling users which is provided by the framework, the whole concept integrates quite well with the Hypertext Model allowing easy personalization of the web-app inner pages. The framework auto-generated application menu can resemble the user current rights by hiding links to the forbidden areas. Another ready made feature is displaying a login page when an unauthorized user tries to directly access a protected page.

Such an approach is quite spectacular for the concepts of web design teaching. Students can virtually see the area “disappearing” from the menu when it gets signed as protected. He/she can get a grasp of web-app structure and inner dependences without the need to first master any serverside language. That lets to conduct more intense parallel training of web application programming and design, both giving the instantly visible results.

Another feature implemented into WebRatio which extends the WebML concept, is the alternative page or area support. A set of alternative areas inside
a page can be modeled in the Hypertext model. The display of the proper area of the set of alternatives can be made dependent on the user credentials, rights, stored parameters or the choices made via adequate links. That can be helpful with teaching a student the concept of the user customizable page design.

Symfony provides complex solutions for access restrictions in web applications i.e. using different database connections for different environments, allowing access to dev environment only to the listed IP addresses, assigning web pages to the users, and many ways of authentication and authorization.

Authentication can be implement basing on tokens or database tables handling user credentials. In the first case modification of submitted content is possible only by using the special token URL (requires routing modifications and writing PHP code) which is generated i.e. using the SHA1 function on e-mail address and random number. In the second case developer is able to create login form on his own or use the sfGuardPlugin [7].

The sfGuardPlugin is an easy to install plug-in that provides authentication and authorization features extending the standard security features of symfony. Unfortunately, during installation, it is necessary to rebuild data model, so all data structures are deleted from data source and created from scratch – another data insertion is required. To secure a symfony application one has to: enable the plug-in, change the default login and secure module, change the parent class in myUser.class.php, add routing rules and mark modules to secure. The sfGuardPlugin comes with three auto-generated modules (forms, widgets, validators, etc.) for managing users, groups and permissions, which is big advantage. All user credentials, including login and profile data, are held in the sfGuarg tables [4].

Before being executed, every action passes through a special filter that checks if the current user has privileges to access the requested action. All security requirements that users must fulfill for each action (content accessing included) are based on user credentials and authentication, and listed in security.yml file in the module ./config directory [7].

The sfGuardPlugin gives advantage to the symfony when it comes to access management and personalization. WebRatio requires a lot of “clicking” and configuring but gives the visualization of access restriction in the WebML hypertext model, letting the students to literally see which pages and operations are secured.

7. LOOK AND FEEL OF THE APPLICATION

Creating the view of the application in the WebRatio is quite a complicated task. The final view is constructed using: standard page layout, units placement
in the grid or custom areas, each unit type layout (with parameters), default pages i.e. the error page.

All that is organized into the style project which can be used by the application. The project can contain several layouts for a considered object. Default layouts can be set but each page and each unit can override that and use the layout of choice (even from the different style project).

Creation of a custom style project requires the knowledge of Groovy Scripts writing and is not aided visually by the framework (Fig. 5).

WebRatio provides a standard style project (called WRDefault) used by all newly created applications. The main disadvantage of this style is the translation of the grid positioning page elements into an outdated and obsolete HTML table design of the page. The more current, layer based style project is not available by default.

The tendency to make web applications more spectacular can nowadays be clearly seen. WebRatio has integrated the AJAX framework called Prototype into the WebRatio AJAX Extension Toolkit. It enables to utilize some standard RIA web behaviors in the applications designed in WebRatio. The Prototype is an opensource framework freely distributable under the terms of a MIT-style license.
The implemented behaviors are: drag & drop, selective page refresh, cascade select management, periodic page refresh (without page reload), dynamic form events management, tooltip, fields auto-completion.

For the designer everything reduces to the enabling of AJAX features and events for the specific units and links. For example in order to cause a link activating the display of a full customer information, not to reload the page, one has to check the Enable AJAX checkbox for the link and the target data display unit (Fig. 6).

![Figure 6. The Hypertext model of partial refreshing the web page](image)

The implemented AJAX features are fully customizable within its scope giving the students an easy way to experience a self created, advanced look of the application with several mouse clicks. On the other hand, an attempt to create the different, custom behavior is a tough task (futile without the mastery of JSP programming) as it would require to manually create or rewrite a part of the application JSP code.

When it comes to symfony, presentation layer can be managed by using layout.php file placed in ./application/templates directory and view.yml configuration file, which are present on all application levels, from global to specific module. Layout templates can be set even for single action like listSuccess (prints list of elements).

Programmer can use view.yml file to set following properties regarding modern look-and-feel of a web application:

- http_metas – i.e. content-type: text/html,
- metas – i.e. title, description, keywords, language, robots,
- stylesheets – paths to CSS files,
- javascripts – paths to JS files (including jQuery),
- layout – name of PHP file with the application layout.

Features given by view.yml are extended in layout.php file (there can be more than one layout.php file), which manages mentioned layout elements and other elements like images, applets and flash animations.

This way of managing presentation layer allows fast switching between layout templates for user and developer. Configuration, involving small number of files, is performed using simple syntax and properly organized. However implementation of very specific non-standardized tasks (i.e. placing the filter text box under the column header or creating cascade of three dynamically reloaded drop-down lists) is more sophisticated and requires framework source code modifications.

When it comes to symfony, common layout creation techniques are fully applicable. Students can freely use AJAX or other RIA techniques, learned prior to the course. Those techniques are not limited by any means and depend only on student’s knowledge. Approach presented by the WebRatio requires the knowledge of Groovy scripting language (in case of creation of a custom style project) therefore an inexperienced developer is limited to the simple predefined layouts only. Also easy to use AJAX features are limited to the small set.

### 8. DEPLOYMENT OF THE APPLICATION

WebRatio testing abilities are much poorer then symfony’s. There is no ability to write unit or functional tests. Low level debugging can be made for custom units, more precisely for unit service classes and requires Java and Eclipse knowledge. At designer level one can use “Find model problems” tool to check data and hypertext models correctness. It shows warnings and errors caused by designer actions, which can be fixed by changing the models.

Applications made using WebRatio require web server handling JSP pages i.e. Apache Tomcat and appropriate database servers. Deployment process is quite simple – preparing and populating database server, manually copying the application files or using deployment plans. Deployment plan is a set of predefined, ordered by developer, activities (tasks) based on deployment configuration settings and Ants tool. Many deployment configurations can be made. Each of them contains path to web server, encoding settings, ports settings etc. One can also specify connection to database and SMTP servers [8].

Middle size application consists of about 1300 files (including 650 files in FCKeditor directory) and requires approximately 40 MB disk space (libraries
size is about 26 MB). There is no problem with overgrowing logs, all required database drivers are included in application files and there is no need to enable them on the server side (advantage against symfony). Unfortunately it is much harder to find hosting server (even not free of charge) than for symfony.

Symfony has well made tools for testing and debugging of the application. One of them – the web debug toolbar (Fig. 7) is present on the top of all web pages in the development environment and shows current application configuration, logs for current request, executed SQL statements, memory and time information [6]. Developers are able to perform two kinds of automated tests – unit and functional tests. Unfortunately writing this tests is time consuming and requires much code writing. However it gives opportunity to provide students with some basics of the tests writing. The documentation for that part of the framework is well prepared. The whole concept can be extended even further to the usage of Selenium IDE giving broader approach to the application testing.

![Timers Table](image)

<table>
<thead>
<tr>
<th>Type</th>
<th>Calls</th>
<th>Time (ms)</th>
<th>Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td>12</td>
<td>13.89</td>
<td>1</td>
</tr>
<tr>
<td>Action &quot;sfGuardAuth\signin&quot;</td>
<td>1</td>
<td>48.71</td>
<td>5</td>
</tr>
<tr>
<td>Yview.&quot;Success&quot; for &quot;sfGuardAuth\signin&quot;</td>
<td>1</td>
<td>74.74</td>
<td>6</td>
</tr>
</tbody>
</table>

Fig. 7. The Web debug toolbar

Symfony provides some tools supporting projects deployment, like project:deploy task which uses SSH and rsync to transfer appropriate files. Excluding above-mentioned, deploying symfony application should be reconsidered as a process consisted of the following steps: framework requirements check, database server preparation, web server configuration, deployment finalization.

Hosting availability (usually mySQL, Apache) is better than for Java, although available ones not always handle sophisticated symfony requirements. There can be troubles with newer or elder PHP engine versions caused by i.e. different way of configuration, depreciated or deleted elements and case sensivity of mySQL server (Linux/Unix vs Windows). Also troublesome, especially for the free hosting servers, is necessity of web server configuration file modification. Besides that, middle size project is consisted of vast number of files (more than 5k) and requires approximately 40 MB disk space, excluding log files, which can reach more than 90MB for each application.
Usage of two different frameworks allows to teach students different approaches of testing and deploying web applications including tests, deployment plans and the role of SSH and rsync in synchronisation of the application files.

9. CONCLUSION

The comparison of frameworks has shown the advantage of the visual design, presented by WebRatio. It is a great aid in the good patterns of the web design teaching, as the student can see the changes in the project and the effect of it. The beginning programmers can quickly grasp the ideas of MVC design, without the need of a prior advanced programming knowledge. On the other hand, that design is limited to the units and features provided by the framework. Any deviation from the set path demands a high level JSP programming knowledge.

When it comes to symfony, the framework requires an intermediate PHP programming knowledge. The visual design is not provided, however the programmer is given a freedom in deciding how to approach the problem of developing specific feature. The intermediate level students can, not only quickly grasp the rules of MVC based design but also extend web application building skills to the enterprise level.

In our opinion, the WebRatio can be used at the beginner level, together with teaching of some server side languages. Such an approach is conducted at the computer science msc courses at the Lublin University of Technology. From the authors experience is clearly shown that grasping the rules of the MVC is not that easy for the students, even if they have finished OOP courses (like C++). As it was shown above, starting the web courses with the WebML lets to fix these problems and switch fluently to web server-side languages and frameworks (like Symfony).

Symfony can then be recommended for the further, advanced teaching of the web application design, where the experienced students can be prepared for the real world, enterprise level application programming.

LITERATURE

1. http://webml.org
Usability comparison of WebRatio and symfony for educational purposes

7. SensioLabs, Symfony: The Definitive Guide to symfony 1.2
8. Web Models, WebRatio User Guide

Manuscript submitted 17.08.2010
Reviewed by Jacek Starzyński, D.Sc., Eng.

WYKORZYSTANIE WEBRATIO I SYMFONY
W PROCESIE NAUCZANIA INŻYNIERII
OPROGRAMOWANIA

Jacek KĘSIK, Kamil ŻYŁA

STRESZCZENIE

Nauczanie inżynierii oprogramowania, wymaga poszukiwania nowych narzędzi, technologii i metodyk oraz sposobów ich wydajnego wykorzystania w kontekście wymagań firm tworzących oprogramowanie.

W artykule dokonano porównania WebRatio i symfony pod kątem ich przydatności w procesie nauczania inżynierii oprogramowania ukierunkowanej na tworzenie aplikacji internetowych zarządzających dużymi ilościami danych, z wykorzystaniem podejścia klasycznego i sterowanego modelami.

Porównanie zostało wzbogacone o spostrzeżenia autorów bazujące na pracy dydaktyczno – naukowej. Dzięki temu artykuł pomaga zaplanować zajęcia tak, aby proces kształcenia spełniał wymogi współczesnego rynku informatycznego.
Jacek KęsiK, Ph.D., Eng. – adiunkt, Computer Science and Numerical Analysis Department, Institute of Computer Science, Lublin University of Technology

Kamil Żyła, M.Sc. Eng. – Junior Assistant Professor, Computer Science and Numerical Analysis Department, Institute of Computer Science, Lublin University of Technology