Framework as a master tool in modern web development

PETR DO, VOJTECH ONDRYHAL
Communication and Information Systems Department
University of Defence
Kounicova 65, Brno, 662 10
CZECH REPUBLIC
petr.do@unob.cz, vojtech.ondryhal@unob.cz

Abstract: The paper presents two web development projects in different timeframes. The focus is on the technology used and architecture applied and also on the speed and quality of product achieved. We would like to discuss the progress that has been made in web development in just five years (at least in our team). We have been working in our department on two web based projects (up to four developers) with comparable content. The first one, Dictionary of military terminology „DICTIONARY“, has started in 2005. The second one, Research portal for university staff „RESEARCH”, has started in 2009. In the first part of the paper, the projects are shortly introduced, than the architecture is discussed and finally the progress in web development that has been made using application frameworks is demonstrated.

Key-Words: Web development, Architecture, UML, PHP, jQuery, CodeIgniter, Framework.

1 Introduction

The main aim of the paper is to demonstrate the progress in web development over last five years we have made in our team. The authors believe that the experience they achieve demonstrates progress in web development in general.

2 Project Description

Follows the description of the projects we have been working on in our department. The projects provide platform for comparisons

2.1 Project DICTIONARY

The project DICTIONARY provides functionality for capturing, editing and publishing terminology in military field. These main functions are extended by supporting modules like discussion forum for individual terms, management of dictionaries, export and import of terms in XML and HTML formats and history of changes.

Four different roles types are defined in this software (reader, contributor, editor and administrator – Figure 1). Reader is an anonymous user who can browse the dictionary, search for terms and read the content of the dictionary. Contributor can discuss the terms and propose new ones. Editor can edit terms and create new terms in his domain. Administrator is a super user who can manage ordinal users, dictionaries, exports and imports, history and the anonymous users' communication (requests for a participation, bug reports, etc.). In a nutshell it is a Wiki type system with managed access.

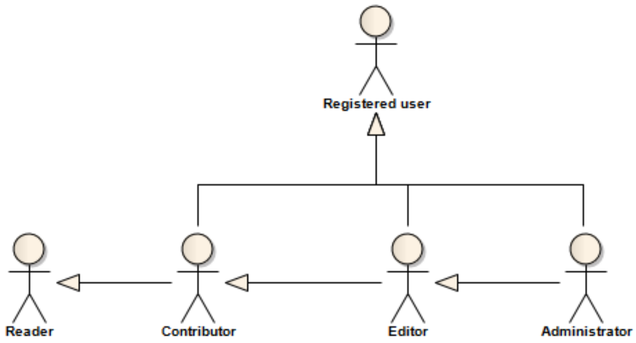
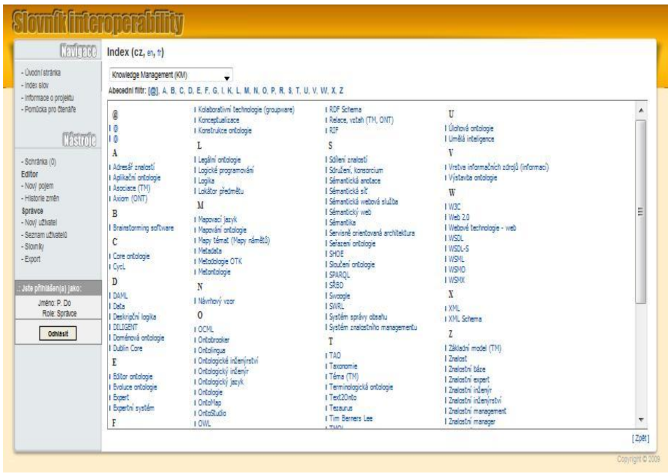


Fig. 1 User roles in DICTIONARY project



Index (cz, en, fr)

Knowledge Management (KM)

Abecední filtr: [@], A, B, C, D, E, F, G, I, K, L, M, N, O, P, R, S, T, U, V, W, X, Z

@
| @
| @

A
| Adresář znalostí
| Aplikační ontologie
| Asociace (TM)
| Axiom (ONT)

B
| Brainstorming software

C
| Core ontologie
| CycL

D
| DAML
| Data
| Deskripční logika
| DILIGENT
| Doménová ontologie
| Dublin Core

E
| Editor ontologie
| Evoluce ontologie
| Expert
| Expertní systém

F

| Kolaborativní technologie (groupware)
| Konceptualizace
| Konstrukce ontologie

L
| Legální ontologie
| Logické programování
| Logika
| Lokátor předmětu

M
| Mapovací jazyk
| Mapování ontologie
| Mapy témat (Mapy námětů)
| Metadata
| Metodologie OTK
| Metontologie

N
| Návrhový vzor

O
| OCML
| Ontobroker
| Ontolingua
| Ontologické inženýrství
| Ontologický inženýr
| Ontologický jazyk
| Ontologie
| OntoMap
| OntoStudio
| OWL

| RDF Schema
| Relace, vztah (TM, ONT)
| RIF

S
| Sdílení znalostí
| Sdružení, konsorcium
| Sémantická anotace
| Sémantická síť
| Sémantická webová služba
| Sémantický web
| Sémantika
| Servisně orientovaná architektura
| Seřazení ontologie
| SHOE
| Sloučení ontologie
| SPARQL
| SŘBD
| Swoogle
| SWRL
| Systém správy obsahu
| Systém znalostního managementu

T
| TAO
| Taxonomie
| Téma (TM)
| Terminologická ontologie
| Text2Onto
| Tezaurus
| Tim Berners Lee

U
| Úlohová ontologie
| Umělá inteligence

V
| Vrstva informačních zdrojů (informací)
| Výstavba ontologie

W
| W3C
| Web 2.0
| Webové technologie - web
| WSDL
| WSDL-S
| WSML
| WSMO
| WSMX

X
| XML
| XML Schema

Z
| Základní model (TM)
| Znalost
| Znalostní báze
| Znalostní expert
| Znalostní inženýr
| Znalostní inženýrství
| Znalostní management
| Znalostní manager

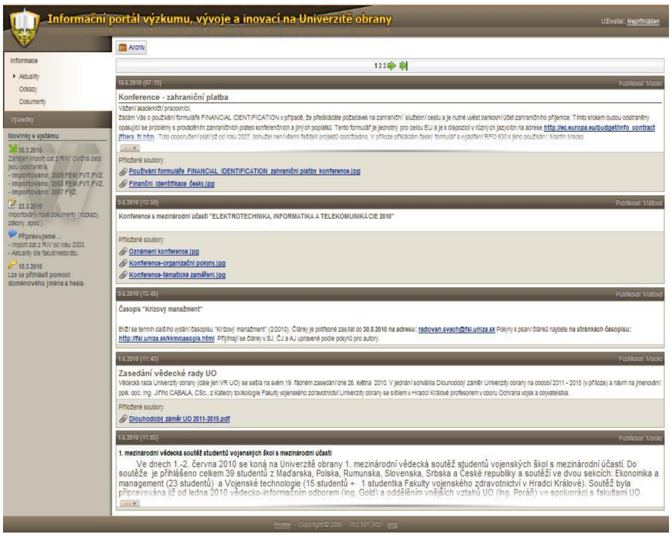
Fig. 2 Main screen of DICTIONARY project

2.2 Project RESEARCH

The RESEARCH project (Figure 3) defines the following functionality:

- Capturing research projects details and members;
- Capturing research outputs in details (18 research output types, each type has unique attributes);
- Providing news in research area to researches;
- Interface to external tools (national level) for research (XML batches for export and import);
- Reporting tools (including graphs) for personal or managers needs.

Only two static user roles are available in the system, user and administrator. Other roles are created dynamically and the functionality assignment to groups or even individuals can be managed.



30.3.2010
Zahájen import dat z RIV. Cvičná data jsou odstraněna.
- Importováno: 2009 FEM,FVT,FVZ.
- Importováno: 2008 FEM,FVT,FVZ.
- Importováno: 2007 FVZ.

22.3.2010
Importovány nové dokumenty (rozkazy, zákony, apod.).

Připravujeme ...
- Import dat z RIV od roku 2003.
- Aktuality dle fakult/rektorátu.

18.3.2010
Lze se přihlásit pomocí doménového jména a hesla.

18.6.2010 (07:19)

Konference - zahraniční platba

Vážení akademičtí pracovníci,
žádám Vás o používání formuláře FINANCIAL IDENTIFICATION v případě, že předkládáte požadavek na zahraniční služební cestu a je nutné uvést bankovní účet zahraničního příjemce. Tímto krokem budou odstraněny opakující se problémy s prováděním zahraničních plateb konferenčních a jiných poplatků. Tento formulář je jednotný pro celou EU a je k dispozici v různých jazycích na adrese http://ec.europa.eu/budget/info_contract/ftiers_fr.htm . Toto doporučení platí již od roku 2007, bohužel není všemi řešiteli projektů dodržováno. V příloze přikládám český formulář a vyjádření RFO 630 k jeho používání. Martin Macko

Přiložené soubory:
Používání formuláře FINANCIAL IDENTIFICATION zahraniční platby konference.jpg
Finanční identifikace česky.jpg

9.6.2010 (13:59)

Konference s mezinárodní účastí "ELEKTROTECHNIKA, INFORMATIKA A TELEKOMUNIKACIE 2010"

Přiložené soubory:
Oznámení konference.jpg
Konference-organizační pokyny.jpg
Konference-tématické zaměření.jpg

9.6.2010 (13:45)

Časopis "Krizový manažment"

Blíží se termín dalšího vydání časopisu "Krizový manažment" (2/2010). Články je potřebné zasílat do 30.8.2010 na adresu: radovan.svach@fsi.uniza.sk Pokyny k psaní článků najdete na stránkách časopisu: http://fsi.uniza.sk/kkm/casopis.html Přijímají se články v SJ, ČJ a AJ upravené podle pokynů pro autory.

1.6.2010 (11:43)

Zasedání vědecké rady UO

Vědecká rada Univerzity obrany (dále jen VR UO) se sešla na svém 19. řádném zasedání dne 26. května 2010. V jednání schválila Dlouhodobý záměr Univerzity obrany na období 2011 - 2015 (v příloze) a návrh na jmenování plk. doc. Ing. Jiřího CABALA, CSc., z Katedry toxikologie Fakulty vojenského zdravotnictví Univerzity obrany se sídlem v Hradci Králové profesorem v oboru Ochrana vojsk a obyvatelstva.

Přiložené soubory:
Dlouhodobý záměr UO 2011-2015.pdf

1.6.2010 (11:03)

1. mezinárodní vědecká soutěž studentů vojenských škol s mezinárodní účastí

Ve dnech 1.-2. června 2010 se koná na Univerzitě obrany 1. mezinárodní vědecká soutěž studentů vojenských škol s mezinárodní účastí. Do soutěže je přihlášeno celkem 39 studentů z Maďarska, Polska, Rumunska, Slovenska, Srbska a České republiky a soutěží ve dvou sekcích: Ekonomika a management (23 studentů) a Vojenské technologie (15 studentů + 1 studentka Fakulty vojenského zdravotnictví v Hradci Králové). Soutěž byla připravována již od ledna 2010 vědecko-informačním odborem (ing. Gold) a oddělením vnějších vztahů UO (Ing. Poráň) ve spolupráci s fakultami UO.

Fig. 3 Main screen of RESEARCH project

3 Development platform

The development platform used in our projects today is still very similar to the platform five years ago, only the versions of the tools and the way tools are used and approaches to development are applied has changed (Figure 4). The projects were modelled using UML language and Enterprise Architect tool.

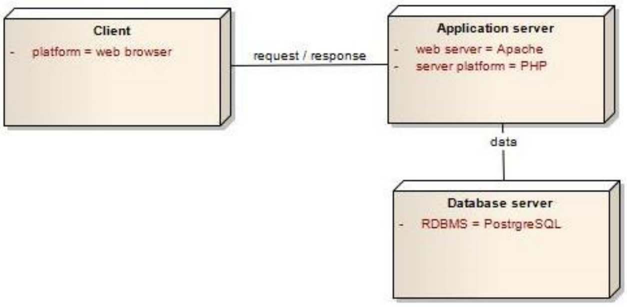


Fig. 4 Development platform

4 Development PROGRESS

In general, the difference between the DICTIONARY and RESEARCH projects is not only in technology utilization, but in project management and team communication. The following general principles were incorporated into RESEARCH project for improving web development:

- The utilization of MVC (Model-View-Controller) design pattern along the whole project; it means both on web client and on web server. Shortly it means data, business and user interface layers separation;
- The object oriented paradigm application;
- Component based development;
- The application of modern tools in web development such as unit testing, application profiling and benchmarking;
- Improvements in configuration and project managements (versioning, change management).

4.1 Client side

Client side is a part of an application the end user is primarily interested in. With technologies evolution, the web browser in more and more better place for making sophisticated user interface. Even so, the traditional desktop client can still achieve more functionality for users, because the thick graphical libraries are included in the operation systems. The stateless HTTP protocol is no more the worst issue. The persistent (state full) connection to the server is maintained at the application level. Cross browser application development is easier with modern browsers, interface and core function differences between browsers are much better balanced and meeting web standards is no more mere vision. The time is here to forget IE6 and the headache it made to developers.

In the DICTIONARY project we have managed all the mentioned issues ourselves. We used only basic JavaScript functionality on the client and standard PHP capabilities on the server. For the RESEARCH project we have decided to utilize application framework; jQuery and jQuery UI JavaScript libraries, but any such libraries would be useful (e.g. prototype and script.aculo.us, etc.). In the following tables we describe the usage of selected technologies (Tables 1, 2).

Table 1 Client technologies for DICTIONARY

Technology utilization	Client - DICTIONARY	
	Yes/No	Description
CSS	Yes	Full utilization.
JavaScript	Partly	Some functions for MD5 support, cookies management or featured text editing.
DOM	No	N/A
External components	Partly	MD5
Web Services	No	N/A
AJAX	No	N/A

Table 2 Client technologies for RESEARCH

Technology utilization	Client - RESEARCH	
	Yes/No	Description
CSS	Yes	Full utilization.
JavaScript	Partly	Including JavaScript jQuery and jQuery UI. Unobtrusive JavaScript approach.
DOM	None	Manipulation of DOM elements with jQuery.
External components	Partly	JQuery plug-ins. CK Editor.
Web Services	None	Google maps (in progress).
AJAX	None	Autocomplete. Clever selecting. Inline editing.

As can be seen from the tables, RESEARCH application is the only one that uses JavaScript language for the development. Developers have changed their meaning of using JavaScript in the past few years. It became a real language for Rich Internet Applications implementation along with AJAX related technologies.

4.2 Server side scripting

As the client side of the application, the server part was improved as well. We still use PHP as a language for a development, but the similarity ends here. The CodeIgniter, PHP application framework, was selected as a server platform for development. The server side application scripts were split into three separate types (Figure 6). The model of traditional approach usually knows only one type of script (the content is usually mixed – Figure 5):

- Controller is responsible for user request processing.
- Model holds data required by a request; it usually communicates with database system for data stored in tables.
- View defines page look (with CSS support).

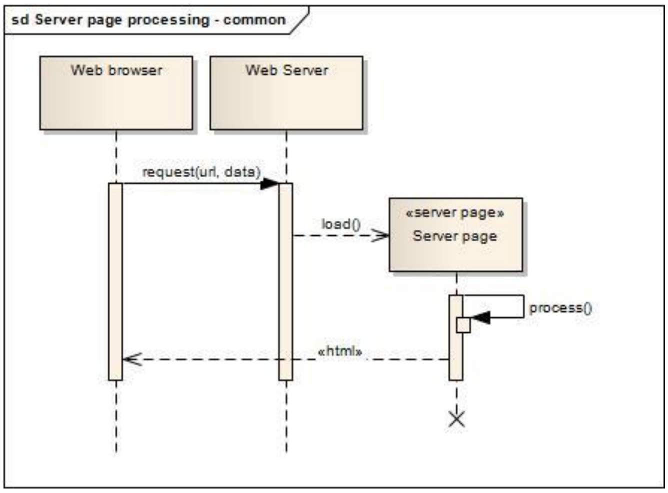


Fig. 5 Server side processing (DICTIONARY)

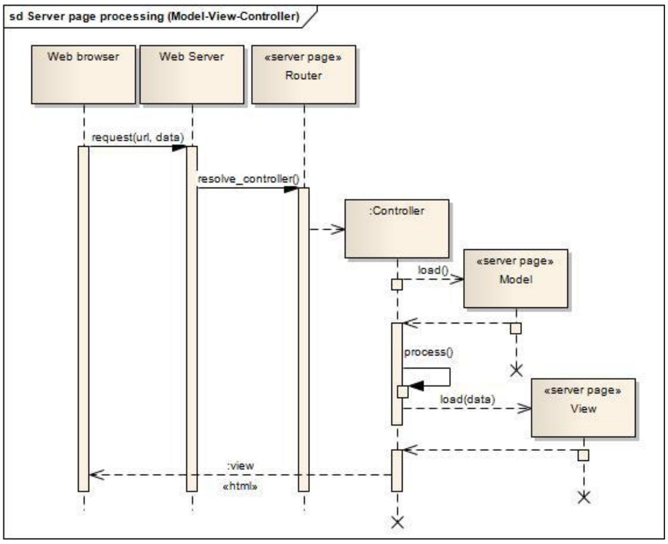


Fig. 6 Server side processing (DICTIONARY)

Table 3 Server technologies for DICTIONARY

Technology utilization	Server - DICTIONARY	
	Yes/No	Description
Objects	No	
MVC	Partly	Separated PHP files for data manipulation and HTML rendering.
External	No	N/A

components		
XML	Yes	DOM.
Objects	No	N/A
MVC	Partly	Separated PHP files for data manipulation and HTML rendering.

Table 4 Server technologies for RESEARCH

Technology utilization	Server - RESEARCH	
	Yes/No	Description
Objects	Yes	When it is suitable.
MVC	Yes	Full adoption with CodeIgniter application framework.
External components	Yes	CodeIgniter plugins. pGraph.
XML	Yes	SimpleXML.
Objects	Yes	When it is suitable.
MVC	Yes	Full adoption with CodeIgniter application framework.

4.3 Client to server communication model

The Web page lifecycle has significantly changed with AJAX technology adoption. In DICTIONARY project any event raised in a web browser ended with page reload (Figure 7). The data passed through the communication channel has to maintain current web page state (e.g. for form’s data validation).

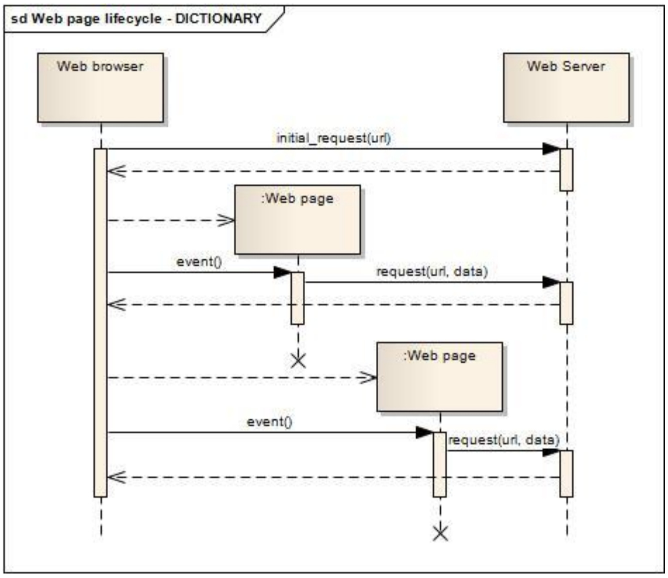


Fig. 7 Web page lifecycle (DICTIONARY)

This has changed in the RESEARCH project. The page is reloaded only when a different functionality is required (Figure 8), so the Web page state persists until another page reload occurs. This improvement is available due to AJAX technology implementation. Any number of asynchronous AJAX calls can be carried out between page reloads.

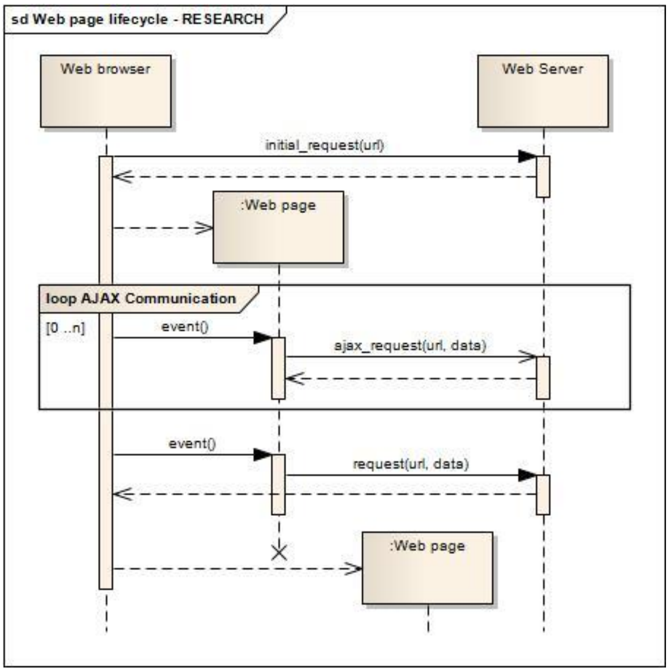


Fig. 8 Web page lifecycle (RESEARCH)

4.4 Data access

Any application incorporates a method in which persistent data is handled. Most of web systems stores data in database systems. Data are stored in relational database management system PostgreSQL in both projects, but MS SQL platform (and a lot of others thanks to robust database layer) is supported for the second project as well. Relational database models (RDM) are created for both applications using Unified modelling language (UML) and Enterprise Architect tool (Tables 5, 6). This part of application development, including database schema generation, is really stable and unchangeable. But the way, the data are accessed in PHP language differs. The abstract data layer (based on Active Record design pattern) provided in application framework allows us to support more target database platforms. Unfortunately the change of the platform was required for merging the RESEARCH system with other university information systems (some data fields are shared and the common university authentication system is adopted) during the development.

The speed of development has significantly increased using the CRUD (Create, Read, Update and Delete) classes. Each table in RESEARCH project has a mirror model class. All such classes are children of common Crud class, where basic methods for database table access are implemented. To access a new table in database system, an “empty” child Crud class is created with all required functionality inherited.

Table 5 Data access for DICTIONARY

Technology utilization	Data - DICTIONARY	
	Yes/No	Description
RDM	Yes	UML. DLL Generation.
Abstract data layer.	No	Code uses native functions for PostgreSQL.
CRUD	No	SQL scripts as required.

Table 6 Data access for RESEARCH

Technology utilization	Data – RESEARCH	
	Yes/No	Description
RDM	Yes	UML. DLL Generation.
Abstract data layer.	Yes	Active record library in CodeIgniter framework.
CRUD	Yes	Each table has associated a class with basic functionality.

4.5 Team communication

The next part compared (Tables 7, 8) is communication between team members and sharing the code. In DICTIONARY project the code sharing and merging was based on folders on the network via FTP and Samba sharing. Releases were managed as folders with appropriate names. Code merging was handled manually. The RESEARCH project started to use versioning control system Subversion. The flexibility of work is improved. Developers can work on the project files in parallel; releases are created automatically in version control system.

Table 7 Team communication for DICTIONARY

Technology utilization	Team - DICTIONARY	
	Yes/No	Description
Model	Yes	UML
Versioning	A little	Manual copy to the new folders.
Change Management	A little	Ad hoc. Bugs first then next features.

Table 8 Team communication for RESEARCH

Technology utilization	Team – RESEARCH	
	Yes/No	Description
Model	Yes	UML
Versioning	Yes	Subversion with email notification.
Change Management	Yes	Bug reporting. Change request management.

4 Conclusion

In the paper, authors shared their experience with web development on two projects in different timeframes. The paper demonstrates the rapid progress that has been made in web development. The new paradigm of frameworks was introduced in the web development for both client and server parts of a web application. This paradigm changed the web development significantly as was demonstrated on the practical examples in the paper. MVC add-on for ASP.NET, which is actually in second version, is only the next confirmation of described web development trends.

Based on our experience on RESEARCH project, it is time for the DICTIONARY project refactoring to improve the comfort of users.

References:

[1] CodeIgniter: An Open source web application framework for PHP. Available [online] at <*http://codeigniter.com/*>.

[2] jQuery: JavaScript library. Available [online] at <*http://jquery.com/*>.

[3] jQuery User Interface: Available [online] at <*http://jqueryui.com/*>.