

# JOHNNY – WEB PLATFORM FOR CREATING INTERNET APPLICATIONS

**Jan Štefl**

Master Degree Programme (2), FIT BUT  
E-mail: xstefl02@stud.fit.vutbr.cz

Supervised by: Tomáš Hruška

E-mail: hruska@fit.vutbr.cz

## ABSTRACT

The vision of this project is to generate source code of a web application based on a model created by the programmer. The programmer will have access to the repository of classes and components, which will be shared with other users of the platform.

## 1. ÚVOD

V průběhu praxe programátora webových aplikací jsem se setkal s problémem neustálého se opakování určitých činností. Dále jsem si všiml, že jedna skutečnost je zapotřebí vždy vyjádřit více způsoby, což zpravidla stojí nemalé úsilí. Tedy například pro nezanedbatelnou část modelovaných entit je třeba vytvářet jak kód generující databázovou tabulku, tak programovou třídu realizující mapování relačních tabulek (*ORM*).

Následující text prezentuje stěžejní rysy vývojové platformy a nastiňuje řešení některých problémů.

## 2. ROZBOR

Východním bodem tvorby aplikace bude její model. Postup převodu na kód se skládá z následujících kroků:

- Vytvoření modelu aplikace pomocí vhodného modelovacího jazyka,
- převod modelu do strojově přívětivé podoby,
- vytvoření interního modelu a
- generování kódu v jazyku cílové platformy.

### 2.1. VYTVOŘENÍ MODELU APLIKACE POMOCÍ VHDNÉHO MODELOVACÍHO JAZYKA

Reálný projekt obsahuje mnoho entit, proto jsem shledal za vhodné použít grafického modelovacího jazyka, který zprostředkovává vyšší perspektivu na daný problém. Konkrétně jsem vybral jazyk UML. Využity jsou modelovací prostředky diagramu případu použití a diagramu tříd. Osobně mám relativně uspokojivé zkušenosti s modelovacím nástrojem *ArgoUML*, který dobře poslouží do doby, než bude vytvořeno webové rozhraní.

## 2.2. PŘEVOD MODELU DO STROJOVĚ PŘÍVĚTIVÉ PODOBY

Pro textovou reprezentaci modelu jsem zvolil jazyk XMI (*XML Metadata Interchange*). Jedná se o popis UML modelu v jazyce XML. Výše uvedený software *ArgoUML* podporuje export do tohoto formátu.

## 2.3. VYTVOŘENÍ INTERNÍHO MODELU

Nyní přichází na řadu aktuálně vyvíjená část platformy. Vlastní implementace je provedena v PHP. Pro načtení a zpracování je použit jazyk XPath s výpomocí DOM.

Interní model reprezentující diagram případu použití je tvořen následující strukturou:

```
$ucDiagrModel[entityId] = array( roleId => operationId )
```

### **Kód 1:** Interní reprezentace diagramu případu použití.

Na základě této části jsou nad jednotlivými entitami tvořeny kontroléry (viz návrhový vzor Model-View-Controller), seznam oprávnění (*ACL*) a vyplývá z něj také URL logika (např.: /entita/operace/id/hodnota).

Interní model reprezentující diagram tříd je tvořen následující strukturou.

```
$classDiagramModel[packageName][className] = array (
    "modifiers" => array( ... ),
    "doc" => array( ... ),
    "parent" => array( ... ),
    "relationWith" => array( ... ),
    "interfaces" => array( ... ),
    "attrs" => array( ... ),
    "methods" => array( ... ),
)
```

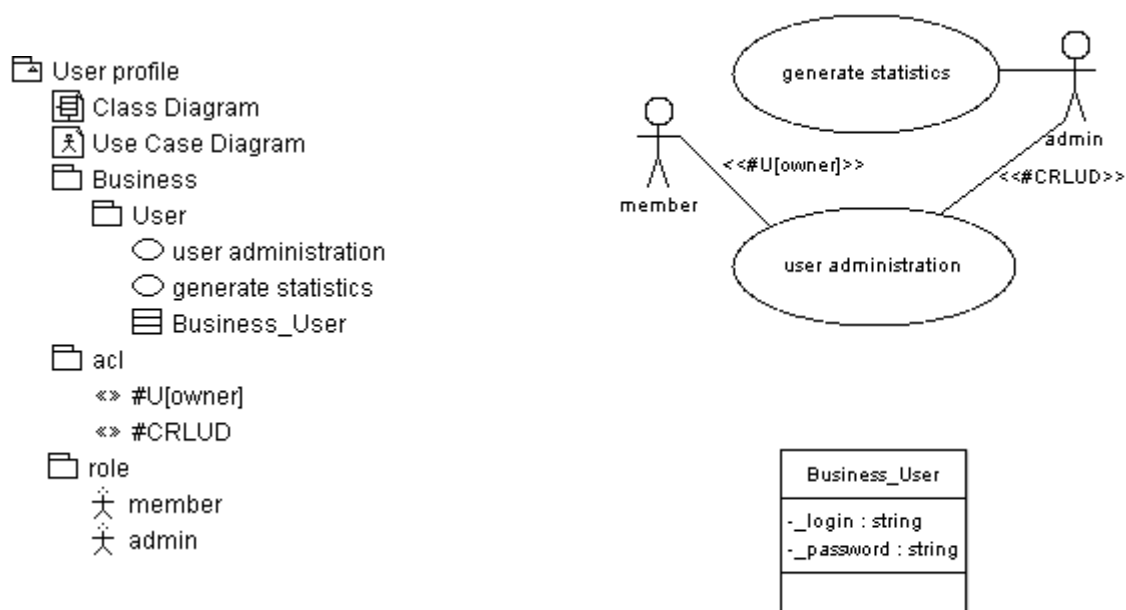
### **Kód 2:** Interní reprezentace diagramu tříd.

Jednotlivé vnořené položky mají odpovídající logiku. Tato část je později transformována na ORM třídy a pro generování databázových tabulek.

## 2.4. GENEROVÁNÍ KÓDU V JAZYKU CÍLOVÉ PLATFORMY

Aktuální cílová platforma je PHP s využitím *Zend Frameworku*. Členění vygenerovaných souborů odpovídá *konvenční modulární adresářové struktuře*, viz [3]. Vlastní kód je v souladu se *Zend kódovými standardy*, viz [4].

## 2.5. PŘÍKLAD



**Obrázek 1:** Modelová struktura, diagram případu použití, diagram tříd.

Obrázek 1 prezentuje základní model vytvořený v *ArgoUML*. Za povšimnutí stojí alternativní zápis v diagramu případu použití. Důvodem zavedení je zmenšení počtu relací mezi rolí a entitou (C = create, R = retrieve, L = list of entites, U = update, D = delete, ...).

## 3. ZÁVĚR

Diskutovaná platforma by měla přinést zefektivnění vývoje webových aplikací. Hlavními činiteli pokroku jsou například automatické generování kódu, sdílený adresář tříd a komponent, a do budoucna kupříkladu reverzní inženýrství, podpora jednotkových testů či implementace konceptu modelem řízené architektury.

I přestože je projekt v raném stádiu vývoje již nyní se dá využít pro generování ORM tříd, viz <http://pis.jan-stefl.eu/>.

## LITERATURA

- [1] Object Management Group: Specification Unified Modeling Language™ (UML®). Dokument dostupný na URL [http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm#UML](http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML).
- [2] Object Management Group: XMI Specification. Dokument dostupný na URL <http://www.omg.org/technology/documents/formal/xmi.htm>.
- [3] Zend Technologies. Using a Conventional Modular Directory Structure. Dokument dostupný na URL <http://framework.zend.com/manual/en/zend.controller.modular.html>.
- [4] Zend Technologies. Zend Framework PHP Coding Standard. Dokument dostupný na URL <http://framework.zend.com/manual/en/coding-standard.html>