Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky
Vyšší odborná škola informačních služeb

**Martin Kukač**

# Návrh interakce školního informačního systému VOŠIS s vybranými internetovými službami

**Bakalářská práce**

2009

vymen za zadavaci list

Prohlašuji, že jsem bakalářskou práci na téma *Návrh interakce školního informačního systému VOŠIS s vybranými internetovými službami* zpracoval samostatně a použil pouze zdrojů, které cituji a uvádím v seznamu použité literatury.

V Praze dne 12. května 2009

# Obsah

## Abstrakt

Tato práce se zabývá možnostmi výstupu dat z informačního systému Vyšší odborné školy informačních služeb do vybraných standardních formátů, jeho napojení na některé internetové služby a ukazuje možnosti jejich implementace. V závěru je zhodnocena úspěšnost této implementace a nastíněna případná úskalí při zavádění do praktického používání.



## Abstract

This paper elaborates on possibilities of data export from VOŠIS information system to standard data formats, possible connection to selected internet services and demonstrates implementation. In the conclusion results of implementation are evaluated and possible difficulties in everyday use are outlined.

# Kapitola 1

# Úvod

Školní informační systém Vyšší odborné školy informačních služeb by stejně jako jakýkoliv jiný v praxi používaný informační systém neměl být uzavřenou jednotkou bez návaznosti na okolní svět. Ačkoliv jsou již dnes studentům nabídnuty některé možnosti, jak pohodlně pracovat s obsahem informačního systému (například zasílání některých informací e-mailem), je značná část informací přístupná pouze z webového rozhraní: přes internetové stránky pro veřejnost, nebo přes intranetové webové rozhraní, které je určeno pouze oprávněným uživatelům, tj. vyučujícím a studentům.

V poslední době je ale trend práce s údaji z informačních systémů jiný. Měnící se obsah webových prezentací je možné sledovat s pomocí syndikačních formátů, nebo vhodným napojením na sociální sítě, kde o nich zainteresovaní uživatelé mohou navíc diskutovat, případně se na ně vzájemně upozorňovat. Pomocí aplikací určených ke správě volného času je možné sledovat online kalendáře a v závislosti na nich si organizovat svoji činnost. Kalendářové a kontaktní informace je možné stahovat přímo z Internetu do mobilních zařízení. Tyto aplikace a zařízení získávají stále větší oblibu i mezi studenty. Umožňují totiž nejen komfortnější práci s informačním obsahem s vynaložením menšího úsilí a času, ale i sjednocení několika informačních zdrojů do jedné aplikace a pohodlnou práci s informacemi, jejich třídění a vizuální prezentaci.

Většina dnes používaných formátů pro výměnu výše nastíněných informací je standardizovaná a tyto specifikace těchto standardů jsou volně dostupné, stejně jako dokumentace k aplikačním rozhraním (API) některých služeb a tak se přímo nabízí možnost jejich napojení na stávající systém.

Tato bakalářská práce se zabývá možností napojení výstupu školního informačního systému na takovéto standardní datové formáty a interakcí školního informačního systému s některými službami dostupnými na síti Internet. Navrhuje jejich implementaci na základě analýzy stávajících datových struktur a demonstruje použitelnost prostřednictvím běžně dostupných aplikací.

# Kapitola 2

# Analýza školního IS

## 2.1 Použité prostředky

Školní informační systém za sebou má poměrně dlouhou historii, důsledkem čehož se jedná o rozmanitou směs nejrůznějších použitých technologií.

Jako programovací jazyk použitý na straně serveru je použito PHP. PHP je skriptovací jazyk, který vznikl v polovině devadesátých let jako malý projekt sloužící k tvorbě dynamických webových stránek a který se postupně vyvinul ve velmi komplexní nástroj, který umožňuje mimo jiné například i objektové programování. Z velkého množství funkci, které je navíc ještě možné dále doplňovat externími knihovnami, stojí za zmínku zejména možnost používat regulární výrazy, zpracování dat z velkého množství databázových serverů nebo práce s grafickými objekty a formáty.

Server, který se přímo stará o generování školních webových stránek používá PHP ve verzi 4, které je dnes již zastaralé, ačkoliv jde stále o plně použitelný skriptovací jazyk. PHP ve verzi 5 je ze školních serverů nainstalováno pouze na testovacím serveru work, na němž byla vyvíjena a testována i většina kódu, který vznikl při tvorbě této bakalářské práce. Zdrojové kódy tedy pro svoji činnost předpokládají právě verzi 5 a vyšší. Nezbytné je to zejména kvůli tomu, že ve verzi 5 bylo PHP obohaceno o některé vlastností, které se úzce dotýkají tématu této práce. Jako jeden z příkladů je možné uvést formát data dle ISO8601, který předchozí verze PHP neobsahovaly a který je použit při generování syndikačního formátu Atom.

Informační systém využívá služeb dvou různých databázových serverů: Informace o vypsaných předmětech, rozvrhu a studentech zapsaných do příslušných kurzů jsou uloženy v databázi Fox Pro, která je pro potřeby PHP připojena přes Open Database Connectivity (ODBC) ovladač, který umožňuje pracovat pomocí jednotného API s různými databázemi, aniž by bylo nutné vědět o jaký typ databáze se jedná. Tento ODBC ovladač, který pracuje přímo se soubory ve formátu DBF, je dostupný pouze na hlavním školním serveru. Testovací server work, na kterém byl vývoj prováděn k němu neměl přístup. Ostatní informace (aktuality, změny rozvrhu, vypsané termíny zkoušek) jsou uloženy na databázovém serveru MySQL, ke kterému lze z PHP přistupovat přímo.

## 2.2 Databáze

Z databáze na MySQL serveru jsou pro cíle této práce využitelné zejména tyto databázové tabulky:

**texty** Tabulka obsahující všechny texty, které se objevují v školních aktualitách, dotazy studentů, oznámení o změnách učeben atd.

**terminy** Tabulka obsahující termíny zkoušek a zápočtových testů.

**termvik** Tabulka s rozvrhem studia dálkového studia.

**prava** Tabulka evidující uživatele s právem zápisu do systému (vyučující, studijní referentky, vedení školy atd.) a základní kontaktní údaje na tyto uživatele.

Tabulky jsou vzájemně jen minimálně provázané a obsahují přímo jména uživatelů, kteří do nich záznam provedli, případně kterých se daný záznam týká. Obecně platí, že není třeba při práci s jednou konkrétní tabulkou vyhledávat data v tabulkách ostatních. Výjimkou je pouze případ, kdy je žádoucí zjistit v souvislosti s záznamem v některé z prvních tří tabulek doplňující údaje ke konkrétnímu uživateli.

Tento přístup není nikterak netradiční a v praxi bývá často upřednostňován před komplikovaným systémem číselníkových tabulek a to zejména v případech, kdy tabulky v databázi mají velký počet řádků. Platí totiž, že i když takové tabulky zabírají mnohem více místa na datovém úložišti, je na jejich zpracování třeba méně procesorového času. Není totiž třeba k nim připojovat příslušné číselníkové tabulky a v nich vyhledávat související údaje, což je i při dobré indexaci vždy úkon, který zabírá procesorový čas. Při současných technických omezeních počítačů platí, že je technicky proveditelnější a ekonomicky výhodnější navyšovat datová úložiště, než výpočetní výkon odpovídajících serverů.

S použitím výše uvedených programovacích a databázových prostředků by mělo být možné realizovat všechny cíle, které si tato práce klade.

# Kapitola 3

# Syndikace aktualit

## 3.1 Vznik a historie syndikačních formátů

S tím, jak World Wide Web začal být v průběhu svého vývoje přístupný pro stále větší skupinu uživatelů, posunul se postupně od původně statických stránek ke stránkám dynamickým. Původně ručně psané stránky, které byly tvořeny lidmi orientujícími se v informačních technologiích a jejichž obsah často nepřibýval ani se neměnil po dobu několika měsíců se změnily ve weby založené na stále přibývajícím a měnícím se obsahu. Tyto stránky, mezi něž patří nejrůznější zpravodajské a informační weby, blogy a e-ziny, jsou generovány sofistikovanými redakčními systémy které umožňují publikovat obsah prakticky každému, kdo ovládá základní práci s počítačem. V důsledku toho došlo postupně nejen k nárůstu obsahu stávajících webů, ale zejména k značnému nárůstu počtu samotných webových stránek a pro odběratele jejich obsahu (tedy čtenáře) začalo být velmi nepřehledné a nesnadné se v takto rychle měnícím prostředí orientovat.

V druhé polovině devadesátých let se proto začaly některé společnosti, pro něž je Internet a zejména web jedním z hlavních oborů působnosti, zabývat řešeními této situace.

Prvním a nejjednodušším způsobem se stalo informování o novém obsahu pomocí e-mailu. Jde o zdánlivě pohodlné řešení, ovšem ukázala se řada nevýhod:

- Každá sledovaná změna obsahu znamená odeslání e-mailu stovkám registrovaných uživatelů, což je rázové zatížení poštovních serverů a datových linek.
- Po roce 2000 došlo k velkému nárůstu nevyžádané elektronické pošty (tzv. spam), mezi kterou mohou tato upozornění snadno uniknout pozornosti uživatele. Některé poštovní servery používají automatické filtrování této nevyžádané pošty a hromadně rozesílané upozornění může být (byť jen dočasně) zařazeno na seznam nevyžádané pošty některého filtru. V důsledku toho pak nedojde až ke koncovému uživateli.
- Případné třídění a archivace těchto upozornění jsou plně na jejich odběrateli. Pokud se rozhodne archivovat si změny na některém serveru a omylem si některé upozornění vymaže (nebo mu nebude z výše uvedených důvodů doručeno), nemá žádný způsob, jak jej opět získat.

Postupem času se do popředí dostaly syndikační formáty. Po několika pokusech o firemní standardy (například Channels od společnosti Microsoft či Meta Content Framework od společnosti Apple) se nejvíce rozšířily standardizované formáty RSS a Atom. Seznamy změn jsou pak odebírány buď pomocí specializovaného software, nebo pomocí standardních aplikací do kterých byla tato funkcionalita přidána. Dnes již podporu zejména RSS obsahují všechny majoritní webové prohlížeče: Microsoft Internet Explorer od verze 7, Mozilla Firefox, Opera od Opera Software i Apple Safari od verze 2.0), existují i webové služby, které umožňují sledování novinek pomocí RSS a Atom odkudkoliv, pouze po přihlášení do webové aplikace (například Google Reader).

Mezi hlavní výhody syndikačních formátů patří:

- Seznam změn je centrálně uložen na serveru, který jej poskytuje klientským aplikacím na jejich požádání. Klientské aplikace žádají pravidelně v zadaných časových intervalech, nebo jednorázově na vyžádání uživatele. To sice znamená absolutně více odeslaných dat než v případě upozornění e-mailem (vyhodnocování toho, co je nové od minulého vyžádání seznamu se děje na straně klientské aplikace, nikoliv na serveru - ten odesílá celý seznam), avšak jde o menší zatížení pro datové linky i servery. Data nejsou odesílána velkému množství odběratelů naráz, ale v průběhu času rovnoměrně jen těm, kteří o to požádají.
- Server může generovat více seznamů najednou. Může tedy existovat například kompletní seznam veškerého obsahu serveru, seznam jen posledních deseti změn, seznamy tříděné po kategoriích článků či seznamy daného autora. Případné třídění a filtrování je tedy na straně klientské aplikace triviální.
- Klientská aplikace navíc může lokálně ukládat obsahy seznamů a tak zpřístupnit odběrateli i historii, která již v aktuálním seznamu na serveru není.

## 3.2 Formát RSS

### 3.2.1 Historie

Za vznikem tohoto formátu stála společnost Netscape, která v roce 1999 představila RSS ve verzi 0.9[18] jako formát splňující sadu specifikací Resource Description Framework (RDF), navrženou konzorciem World Wide Web Consortium (W3C) jako datový model pro metadata. Vzápětí byla ale vytvořena varianta označovaná jako verze 0.91, která formát zjednodušila a odstranila z něj všechny RDF elementy. Zde došlo k rozdělení RSS na dvě větve:

První větev je odvozená od původního návrhu (0.9), kde RSS je zkratka slov RDF Site Summary. Za účelem rozšíření původního formátu byla vytvořena pracovní skupina RSS-DEV, mezi jejímiž členy byl například Ramanathan V. Guha, který původně vyvíjel

Meta Content Framework pro Apple a později byl klíčovou osobou ve vývoji RSS 0.9. Tato pracovní skupina v roce 2000 prezentovala standard RSS 1.0, který byl stejně jako původní verze odvozen od RDF modelu. Tato verze (která je nekompatibilní se všemi ostatními RSS verzemi) zavádí použití jmenných prostorů XML[1], díky nimž je možné používat libovolné XML elementy, které použitý jmenný prostor definuje. To prakticky znamená možnost nejrůznějšího specifického využití jinak obecného formátu a zároveň i jeho vývoj bez nutnosti vydávání nových verzí. Software zpracovávající RSS kanál pak může elementy z pro něj neznámých jmenných prostorů ignorovat. Nicméně účelem formátu RDF Site Summary, jak název ostatně napovídá, bylo vytvořit spíše souhrn obsahu webových stránek, než zajišťovat syndikaci. Proto se masivnímu rozvoji dostalo zejména druhé vývojové větvi a tato, založená na RDF je nyní bez dalšího vývoje.

Druhá větev je odvozená od verze 0.91, RSS je zkratka slov Rich Site Summary, respektive Really Simple Syndication. Společnost UserLand Software, která tuto revizi vytvořila ji prohlásila za své vlastnictví a její další vývoj byl tedy pouze v režii této společnosti. Z této větve jsou odvozeny verze RSS 0.92, RSS 2.0 a nerealizované návrhy RSS 0.93 a RSS 0.94. Verze 2.0 uvedená v roce 2003 je zatím poslední existující specifikací tohoto formátu. Vývoj byl zmrazen a autoři připouští další verze pouze za účelem zpřesnění některých pojmů, nikoliv za účelem rozšiřování vlastností. Verze 2.0 také zavádí použití jmenných prostorů XML, se všemi výhodami z toho plynoucími. RSS 2.0 je zpětně kompatibilní s předchozími publikovanými standardy stejné větve, tj. dokumenty které splňují specifikaci RSS 0.91 a RSS 0.92 jsou i validními RSS 2.0 dokumenty. V červnu roku 2003 byla práva související s RSS 2.0 převedena na Berkman Center & Internet and Society a zároveň vznikla poradní rada pro vývoj RSS (RSS Advisory Board), která řeší případné pochybnosti ve specifikaci a vydává zpřesňující verze. Současná zpřesňující verze má označení 2.0.11[17].

### 3.2.2 Popis formátu

Pro implementaci v školním informačním systému byl zvolen formát RSS 2.0 ve verzi 2.0.11. Jedná se o variantu jazyka XML, všechny RSS dokumenty musí splňovat specifikaci XML 1.0 podle W3C. Soubor začíná XML deklarací, ve které nachází verze XML, případně národní kódování, ve kterém je dokument publikován. Samotný obsah dokumentu se nachází uvnitř tagu <rss>, který má parametr označující použitou verzi standardu. Každý soubor obsahuje právě jeden element <channel>, který obsahuje tzv. channel elements, ve kterých jsou uvedené informace o samotném kanálu a o webu k němuž se kanál vztahuje. Dále může obsahovat samotné informace o obsahu, respektive o jeho změnách. Tyto informace jsou obsaženy v elementech <item>, kterých může být libovolný počet. Informace jsou strukturovány v tzv. item elements.

[1] XML namespaces, viz. http://www.w3.org/TR/REC-xml-names/

**Channel elements** Dělí se na povinné elementy, které musí kanál obsahovat vždy a volitelné elementy, které obsahují dodatečné informace. Volitelných elementů je relativně velké množství, uvádím tedy jen ty, které jsou použity v implementaci.

Povinné elementy:

**<title>** Titulek RSS kanálu. Specifikací je doporučeno, aby byl stejný jako titulek webové stránky, které se týká.

**<link>** URI webové stránky které se RSS kanál týká.

**<description>** Stručný popis obsahu kanálu, respektive webové stránky.

Volitelné elementy:

**<language>** Jazyk obsahu kanálu. Jazyk je tvořen pěti znaky ve formátu "xx-YY", jejichž význam je odvozen od RFC1766[21]. První část zapisovaná malými písmeny označuje jazyk podle normy ISO639. Pomlčka a druhá část zapisovaná velkými písmeny je nepovinná a označuje jazyka podle ISO3166. Například tedy americká angličtina má kód ve tvaru "en-US", čeština má pak označení "cs-CZ".

**<copyright>** Obsahuje informaci o autorských právech obsahu kanálu.

**<webMaster>** Kontakt na osobu technicky zajišťující provoz RSS kanálu. Vhodné pro kontakt v případě nekorektního generování kanálu. Kontaktem je myšlena e-mailová adresa, kterou tento tag musí obsahovat, jinak není vyhodnocen oficiálním RSS validátorem jako validní. Jméno osoby se může nacházet v kulatých závorkách za emailem, např.: "skopec@sks.cz (Skopec Antonín)"

**<item>** Element obsahující jednotlivé příspěvky. Každý příspěvek musí mít svůj element <item>

**Item elements** Všechny elementy spadající které jsou sdruženy v elementu <item>jsou volitelné, avšak platí, že musí být obsažen alespoň jeden element z dvojice <description>a <title>:

**<title>** Titulek příspěvku.

**<link>** URI odkazující na plné znění příspěvku.

**<description>** Synopse příspěvku, případně jeho plné znění.

**<author>** Autor příspěvku, formát tohoto elementu je stejný jako v případě elementu webMaster u hlavičkových elementů.

**<guid>** Jedinečný identifikátor příspěvku. Může se jednat o URI použité v elementu link, pokud je zaručena jeho jedinečnost.

**<pubDate>** Datum publikování příspěvku. Datum musí být ve formátu podle RFC822, tj. například `Tue,`05 May 2009 00:00:00 +0200

**<category>** Kategorie do které příspěvek spadá.

### 3.2.3 Implementace

Syndikační formáty jsou ideálním nástrojem, který umožní pohodlné sledování příspěvků v sekci Aktuálně na webových stránkách VOŠIS. V této sekci se objevují zprávy od vyučujících pro studenty, změny v rozvrhu atd. Vzhledem k tomu, že v této sekci přibývá několik příspěvků denně, je žádoucí mít možnost filtrovat její obsah. V současné době generuje informační systém pro tuto sekci RSS kanál, který ovšem není na straně serveru filtrovatelný. Student, který ho používá tak odebírá všechny aktuality, včetně těch které se netýkají jeho studia. Prvním krokem je tedy vytvoření RSS kanálu, který tento problém vyřeší.

Při implementaci bylo nejdříve nutné rozhodnout se, zda se generátor RSS kanálu pojme jako jednorázový skript, nebo se vytvoří univerzálnější řešení, které půjde použít i pro případné další formáty syndikace. Jelikož je součástí této práce i implementace generátoru syndikačního formátu Atom, byla koncepce jednorázového skriptu zavržena a vznikl šablonový systém, který je možné při minimálních úpravách použít prakticky pro jakýkoliv formát, který se stejně jako typický RSS dokument se skládá ze tří částí, které je možné brát jako uzavřené celky:

1. hlavička, obsahující začátek obsahu popis kanálu (v případě RSS jsou to channel elements)

2. obsah tvořený příspěvky stejné struktury (zde opakující se element <item>¿ a item elements)

3. patka (zakončení elementů otevřených v hlavičce)

Vznikly tedy tři šablony: `rss_header.inc` pro hlavičku, `rss_item.inc` pro příspěvky a `rss_footer.inc` pro patku. Obsahují přímo XML kód z nějž se ve výsledku skládá RSS dokument. Na místech, která mají být nahrazena obsahem jsou klíčová slova uzavřená do složených závorek. Složené závorky byly vybrány jako speciální znak, který se nikde v RSS dokumentu nevyskytuje a proto jednoznačně odděluje klíčové slovo od samotného XML kódu. Každé klíčové slovo má definován obsah, kterým je nahrazen - pro channel elements jsou to stejnojmenné konstanty definované v konfiguračním souboru (config.php) a pro item elements jsou to odpovídající databázová pole, respektive jejich formátovaný obsah. Samotný skript pak prochází šablony nahrazuje v nich odpovídající pole a výstup odesílá jako RSS kanál (MIME typ application/rss+xml). Změna struktury RSS dokumentu tak spočívá pouze ve změně šablony, zásah do skriptu vyžaduje až změna plnění obsahem.

Ukázka z šablony rss_header.inc

```
<title>{TITLE}</title>
<link>{LINK}</link>
```

Definice konstant v config.php

```
define (TITLE, "VOŠIS");
define (LINK, "http://www.sks.cz/");
```

Odpovídající části skriptu rss.php

```
$text = str_replace("{TITLE}", TITLE, $text);
$text = str_replace("{LINK}", LINK, $text);
```

Výsledek po zpracování sktiptem:

```
<title>VOŠIS</title>
<link>http://www.sks.cz/</link>
```

Generátoru RSS kanálu lze předat několik parametrů, jimiž lze již na straně serveru provádět filtrování obsahu kanálu:

**autor** Filtr podle autora příspěvku, tj. vyučujícího, studijní referentky atd. Jméno se zadává s diakritikou, lze zadat i jen část jména či příjmení. Pokud je zadáno celé jméno, musí být ve tvaru "Příjmení Jméno", jinak nebude autor nalezen. Zde je třeba mít na paměti, že některé příspěvky nemají autora (například změny v rozvrhu) a těm je přidělen výchozí autor VOŠIS.

**predmet** Filtr podle předmětu, kterého se příspěvek týká. Předmět se zadává pomocí kódu, který má v systému školy

**limit** Počet příspěvků, které v RSS kanálu budou. V případě nezadání je počet omezen podle konstanty ITEMS definované v konfiguračním souboru RSS generátoru

Aby nebylo nutné seznamovat se při používání RSS kanálu s těmito parametry, byl vytvořen skript, který je přímo z databázové tabulky obsahující aktuality schopen vygenerovat seznam všech autorů, kteří někdy zadali aktualitu a seznam všech předmětů, pro které byla někdy aktualita zadána. Každý kanál má možnost omezení na 20, 50 a 100 příspěvků nebo bez omezení (tj. bude použita hodnota ITEMS). Tento seznam je prezentován v podobě HTML stránky.

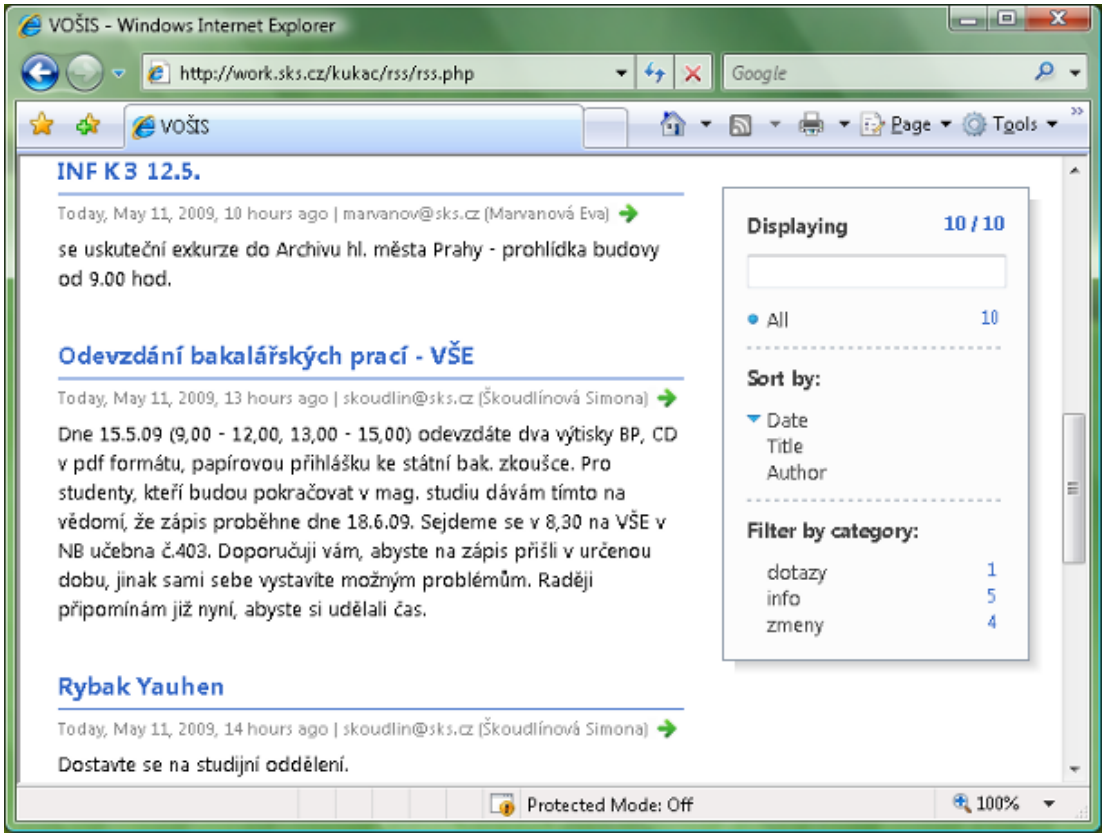
### 3.2.4 Ukázka RSS dokumentu

Následující ukázka kódu je část RSS dokumentu generovaného skriptem popsaným v předchozí kapitole. Jedná se o validní RSS ve verzi 2.0.

```
<?xml version="1.0" encoding="Windows-1250"?>
<rss version="2.0">
 <channel>
  <title>VOŠIS</title>
  <link>http://www.sks.cz/</link>
  <description>Vyšší odborná škola informačních služeb.</description>
  <language>cs-CZ</language>
  <copyright>Copyright by Vyšší odborná škola informačních služeb.
  </copyright>
  <webMaster>skopec@sks.cz (Skopec Antonín)</webMaster>
     <item>
   <title>Přesun konzultačních hodin</title>
   <guid>http://www.sks.cz/odkazy/rss.phtml?id=8192</guid>
   <link>http://www.sks.cz/odkazy/rss.phtml?id=8192</link>
   <description>Konzultační hodiny ze 7. 2.
      přesouvám na 13. 2. od 10,30 do 12 hodin.</description>
   <author>simek@sks.cz (Šimek Eduard)</author>
   <pubDate>Fri, 02 Jan 2009 12:34:56 +0200</pubDate>
   <category>info</category>
  </item>
 </channel>
</rss>
```

### 3.2.5 Příklad použití

V prostředí Windows XP nebo Windows Vista lze RSS kanál odebírat pomocí integrované čtečky ve webovém prohlížeči Internet Explorer 7. Prohlížeč sám detekuje z kanálu kategorie a umožňuje podle nich aktuality třídit.




Obrázek 3.1: RSS kanál odebíraný integrovanou čtečkou v Internet Exploreru.

## 3.3 Formát Atom

### 3.3.1 Historie

Atom je ve svých principech velmi podobný formátu RSS. Také je založen na jazyce XML a jeho primárním účelem je syndikace obsahu. Vznikl v roce 2003, tedy již v době, kdy bylo RSS přivedeno do verze 2.0 a jeho další vývoj byl zmrazen. Právě to vedlo k tomu, že nespokojení uživatelé tohoto formátu, kteří chtěli přidávat novou funkčnost, nebo jen pojmout řešení problému syndikace jinak, začali navrhovat formát vlastní[19]. Původně se jednalo o zcela veřejnou diskusi, která vedla až k prvním specifikacím formátu Atom (verze 0.2 a 0.3). Verze 0.3 již začala být hojně používána, jeho prosazení ale zaštítila společnost Google. Skupina vývojářů, která Atom na základě diskuse publikovala se rozhodla standardizovat ho a ve spolupráci s IETF[2] tak vznikla v roce 2004 pracovní skupina

[2]Internet Engineering Task Force (asociace vytvářející internetové standardy)

Atompub. V následujících letech vznikly navrhované standardy RFC4287[15], který popisuje samotný syndikační formát Atom, a dále RFC5023[16], který popisuje publikační protokol pro tento formát.

Mezi výhody formátu Atom proti formátu RSS patří zejména jasné deklarování typu obsahu. V RSS mohou elementy obsahovat pouze čistý text nebo HTML (v případě použití HTML musí být všechny nepovolené nealfanumerické znaky převedeny na tzv. escape sekvence, tj. nahrazeny speciálními kódy) a není žádný způsob jak označit, který způsob byl použit. Jednotlivé elementy dokumentu ve formátu Atom mají typ obsahu je vždy explicitně deklarován a klient jej může vynechat, neumí-li jej zpracovat. Obsahem elementů může být některý z následujících prvků:

- čistý text
- HTML (nepovolené znaky převedené na escape sekvence)
- XHTML
- XML
- binární obsah zakódovaný metodou base64
- ukazatel na webový obsah mimo samotný dokument

Další vlastností Atomu je oddělení stručného souhrnu obsahu příspěvku od jeho plné délky. U RSS není nijak označeno, obsahuje-li kanál příspěvky v plném znění, nebo nikoliv. Atom zavádí element <summary>, který slouží k stručnému popisu příspěvku (tj. úryvek nebo textový popis netextového obsahu) a element <content>, který může obsahovat celý příspěvek. V neposlední řadě je výhodou, že Atom sám je jmenným prostorem XML, zatím co RSS může sice obsahovat elementy různých jmenných prostorů, ale formát sám o sobě jmenným prostorem není. Existuje mnoho dalších vlastností, které Atom odlišují od RSS, avšak při běžném sledování změn na webových stránkách nejsou tolik podstatné. Atom je totiž využíván jako datový formát pro API některých webových služeb (například Google Data API) a další vlastnosti jsou cíleny zejména na toto použití.

### 3.3.2 Popis formátu

Dokument (označovaný jako Atom feed) má část, která popisuje feed a webové stránky, ke kterým se váže (feed elements) a pak část, ve které jsou samotné příspěvky (entry elements). Na rozdíl od RSS zde ale obě části mají povinné elementy. Dokumentace pro vývojáře navíc uvádí i doporučené elementy, které sice nejsou povinné, ale zajistí optimálně použitelný výstup. V následujícím seznamu jsou uvedeny pouze implementované volitelné elementy, neboť seznam všech volitelných elementů přesahuje rozsah této práce.

**Feed elements**

Povinné elementy

**<id>** Identifikace pomocí jednoznačného a stálého URI. Vývojářská dokumentace uvádí, že optimálním obsahem tohoto elementu je doménové jméno webu, kterého se feed týká.

**<title>** Titulek Atom feedu. Dle dokumentace by mělo jít o pro člověka srozumitelný titulek, doporučuje se použít stejný titulek, jaký má webová stránka, k níž se feed váže

**<updated>** Datum poslední aktualizace feedu ve formátu ISO8601.

Doporučené elementy

**<author>** Feed musí obsahovat element <author>, pokud tento element neobsahují všechny příspěvky (tj. všechny elementy <entry>). Je proto doporučeno element zahrnout vždy. Obsahuje povinně element <name>se jménem autora a nepovinně dva elementy <email >a <uri>ve kterých se nachází e-mailová adresa autora, respektive adresa jeho webových stránek.

**<link>** URI odkazující na stránku, ke které se daný feed váže. Element má celou škálu parametrů, pomocí nichž je možné odkázat se na různé jazykové verze webové stránky, případně na různé její verze (verze v HTML, verze v čistém textu atd.). Dle doporučení autorů by feed měl obsahovat odkaz sám na sebe, aby v případě jeho uložení bylo možné identifikovat původní zdroj.

Volitelné elementy

**<subtitle>** Podtitulek stránky, může obsahovat doplňující vysvětlení k feedu.

**Entry elements**

Povinné elementy

**<id>** Identifikace příspěvku pomocí jednoznačného a stálého URI. Dva příspěvky v rámci jednoho feedu mohou mít stejné ID, pouze pokud se jedná o dvě různé časové verze stejného příspěvku (například odkaz na původní a aktualizovanou verzi dokumentu).

**<title>** Titulek příspěvku.

**<updated>** Datum poslední aktualizace příspěvku ve formátu ISO8601. Dle dokumentace není nutné měnit tuto položku při drobných změnách v obsahu příspěvku, jako je například oprava pravopisné chyby.

Doporučené elementy:

**<author>** Každý příspěvek musí obsahovat alespoň jeden element <author>, pokud jej neobsahuje nadřazený element <feed>.

**<content>** Obsahuje kompletní obsah příspěvku, nebo odkaz na něj. Pokud není součástí příspěvku odkaz na alternativní verzi příspěvku, musí být element vyplněn.

**<link>** URI stránky, ke které se příspěvek váže. Platí stejná pravidla jako u stejnojmenného elementu ve feed elements.

**<summary>** Stručný souhrn příspěvku. Měl by být vyplněn v případě, že element <content>obsahuje netextová data, nebo v případě, že element content chybí úplně (Při implementaci byl tento element vynechán, neboť tyto podmínky nejsou splněny).

### 3.3.3 Implementace

V případě Atom feedu byl znovu použit skript implementující RSS kanál. Byly pouze vytvořeny nové šablony: `atom_header.inc`, `atom_item.inc` a `atom_footer.inc`. Změny v samotném skriptu generujícím soubor s Atom feedem byly spíše kosmetické. Po zpracování šablon je výsledek odeslán jako Atom feed (MIME typ application/atom+xml). Skript přijímá stejné parametry, jako skript generující RSS a díky tomu lze i Atom filtrovat na straně serveru. Zároveň byl Atom přidán do souhrnné stránky, která obsahuje seznamy autorů, předmětů a filtry na počet příspěvků.

### 3.3.4 Ukázka Atom dokumentu

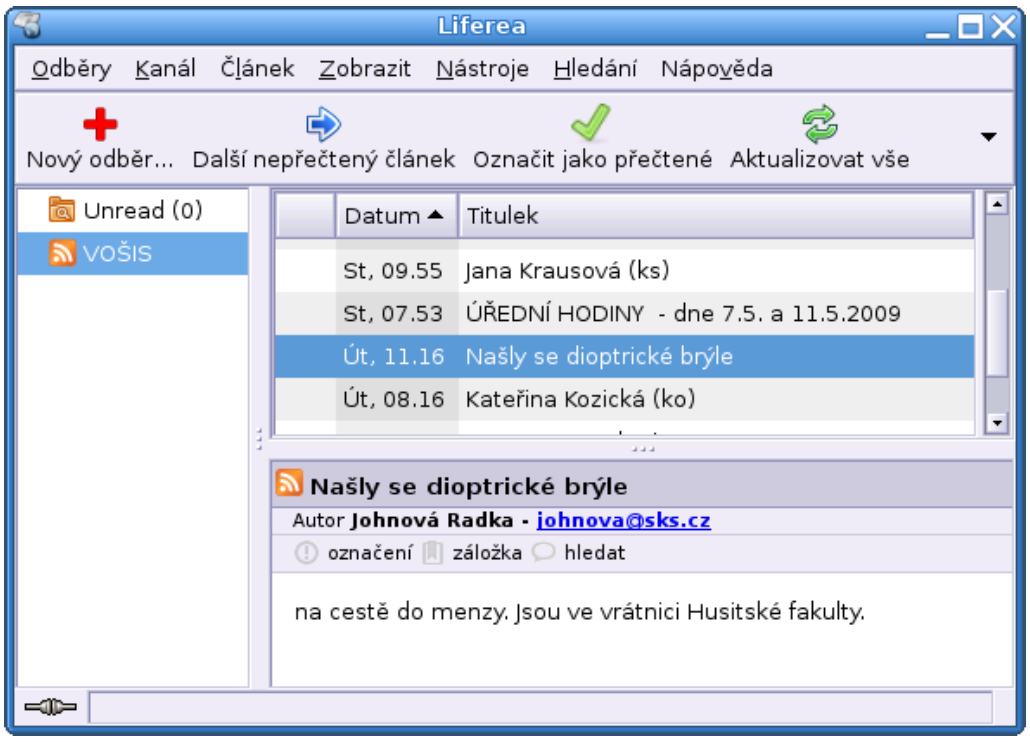
Následující ukázka je část Atom dokumentu generovaného skriptem popsaným v předchozí části. Jedná se o validní kód ve formátu Atom ve verzi 1.0.

```
<?xml version="1.0" encoding="Windows-1250"?>
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="cs">
 <title>VOŠIS</title>
 <subtitle>Vyšší odborná škola informačních služeb.</subtitle>
 <id>http://www.sks.cz/</id>
 <link href="http://www.sks.cz/" />
 <link rel="self" href="http://work.sks.cz/kukac/atom/atom.php" />
 <updated>2009-05-05T01:23:45+01:00</updated>
 <author>
  <name>VOŠIS</name>
  <email>info@sks.cz</email>
 </author>
```

```
<entry>
  <title>Unix bez výuky</title>
  <link href="http://www.sks.cz/odkazy/rss.phtml?id=10908"></link>
  <id>http://www.sks.cz/odkazy/rss.phtml?id=10908</id>
  <content>předtermín bude 4.5 2009 5 a 6 hodina, učebna 1.3.</content>
  <author>
   <name>Klimánek David</name>
   <email>klimanek@sks.cz</email>
  </author>
  <updated>2009-04-14T12:34:56+01:00</updated>
 </entry>
</feed>
```

### 3.3.5 Příklad použití

V operačním systému Linux lze změny obsahu syndikovat mj. pomocí specializované aplikace Liferea, která kromě formátu Atom umí pracovat i s formátem RSS.




Obrázek 3.2: Atom feed odebíraný čtečkou Liferea.

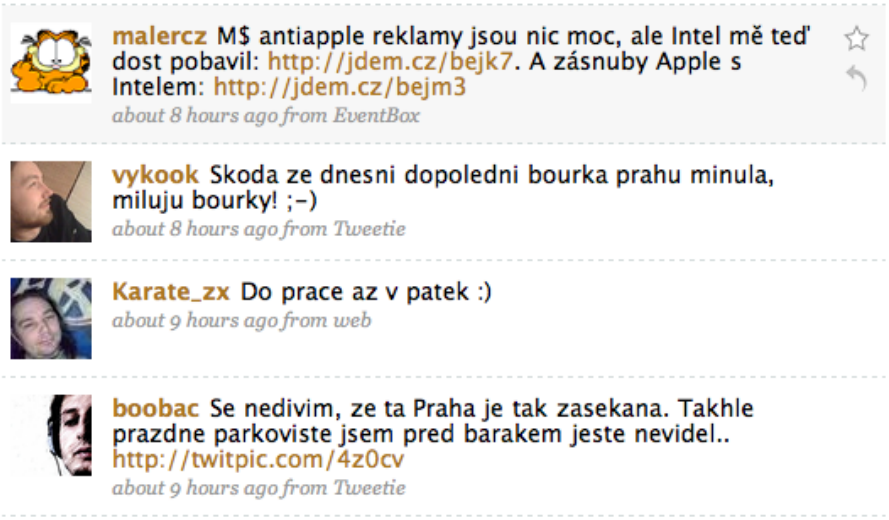
# Kapitola 4

# Interakce se službami sociálních sítí

## 4.1 Sociální síť Twitter

### 4.1.1 Vznik a historie sítě Twitter

Twitter je služba kombinující vlastnosti mikroblogu a sociální sítě. Vznikla v roce 2006 a umožňuje registrovaným uživatelům publikovat krátké příspěvky (maximálně 140 znaků). Primárním obsahem těchto příspěvků má být popis činnosti, kterou daný uživatel právě vykonává (tzv. stavová zpráva nebo jen status). Z těchto zpráv je pak chronologicky generováno několik výpisů. Předně je to výpis veřejný (pokud uživatel nezakázal zveřejňování svého statusu), který obsahuje stavové zprávy všech registrovaných uživatelů, dále pak výpis uživatelský, přístupný pouze po přihlášení k účtu. Tento seznam shrnuje všechny zprávy uživatele samotného a všech jeho přátel, které si přidal do seznamu a kteří mu povolili odebírat jejich stavové zprávy.




Obrázek 4.1: Příklad uživatelského výpisu stavových zpráv.

Jedná se o jednu z nejpoužívanějších sociálních sítí na světě vůbec a proto logicky došlo k jejímu využívání i za komerčními, politickými a obecně informačními účely. Velké korporace jako Apple či IBM využívají Twitter k propagaci nových produktů, nebo firemních akcí. Politikům slouží jako informační nástroj během volební kampaně i mimo ni - svůj profil na Twitteru má například americký prezident Barack Obama či občanská kampaň za zvolení Ivety Radičové prezidentkou Slovenské republiky v prezidentských volbách v roce 2009. Zejména ovšem na Twitteru můžeme najít profily značné části internetových magazínů a blogů, kterým slouží k upozorňování na nově publikované články (v České

republice například internetový IT magazín Živě, blog Cuketka.cz a další). Proto se logicky jeví jako další vhodná možnost upozorňování na aktuality publikované ve školním intranetovém systému.

### 4.1.2 Twitter API

Twitter nabízí veřejně přístupné aplikační rozhraní (API)[22], které umožňuje kompletní obsluhu této služby: publikování nových stavových zpráv i odebírání veřejného či soukromého seznamu. Jedná se v principu o rozhraní velmi jednoduché, splňující specifikaci REST[1][7], tj. jde o bezestavovou komunikaci typu klient-server, založenou na HTTP protokolu. Komunikace spočívá v odesílání zpráv klientem metodou POST na URI zdroje. K ověření uživatele je na výběr otevřený ověřovací protokol Open Authentication (OAuth), jehož podpora ze strany Twitteru byla v době psaní této práce pouze ve stádiu betaverze, nebo základní metoda HTTP Basic Authentication. Odpověď serveru se skládá z HTTP stavového kódu a samotné zprávy v požadovaném formátu, která je získána metodou GET.

Ze stavových kódů mají praktický význam tyto kódy:

**200 OK** Požadavek byl úspěšně vyřízen.
**400 Bad Request** Chyba v požadavku na straně klienta. Zpráva obsahuje popis chyby.
**401 Not Authorized** Chyba v přihlašovacím jménu a heslu.
**403 Forbidden** Server odmítá vyřízení požadavku. Zpráva obsahuje popis chyby.
**404 Not Found** Požadavek na neexistující URI, nebo neexistuje uživatel pod jehož jménem se požadavek provádí.
**500 Internal Server Error** Nespecifikovaná chyba na straně Twitter serveru.
**502 Bad Gateway** Server je právě v režimu údržby nebo nedostupný.
**503 Service Unavailable** Server je dostupný, ale nezvládá zpracovat množství požadavků.

Zpráva samotná pak může být v některém z následujících formátů:

**RSS** Syndikační formát RSS ve verzi 2.0.

**Atom** Syndikační formát Atom ve verzi 1.0.

**XML** Jednoduché strukturované XML, podle specifikace XML 1.0 od W3C. Následuje ukázka části zprávy v tomto formátu:

```
<status>
  <created_at>Sun May 10 20:14:55 +0000 2009</created_at>
  <text>Prave ze spektristi jsou na tom nejhure.</text>
```

[1] Representational State Transfer

```
  <user>
    <name>Sweet</name>
    <location>Pilsen</location>
  </user>
</status>
```

**JSON** JavaScript Object Notification - formát shodný se zápisem definice objektů v JavaScriptu podle RFC4627. Následující ukázka ve formátu JSON obsahuje část stejné zprávy, jako ukázka předchozí:

```
{
"text":"Prave ze spektristi jsou na tom nejhure.",
"created_at":"Sun May 10 20:14:55 +0000 2009",
"user":{
  "name":"Sweet",
  "location":"Pilsen"
  }
}
```

Formáty RSS a Atom jsou použitelné pouze pro požadavky, jejichž výstupem je veřejný nebo uživatelský seznam stavových zpráv. V případě, že klient použije tyto formáty pro komunikaci s jiným zdrojem (například při pokusu o zaslání nové stavové zprávy), odpoví server chybou. Formáty XML a JSON jsou proti tomu univerzální, server v nich umí vrátit seznamy stavových zpráv, popisy chyb a jakoukoliv další komunikaci.URI na které se požadavek odesílá má tvar http://twitter.com/statuses/zdroj.format.

Základní zdroje nabízené Twitterem:

**public_timeline** Veřejný seznam stavových zpráv

**friends_timeline** Seznam stavových zpráv daného uživatele a uživatelů, které má ve svém seznamu přátel

**user_timeline** Seznam stavových zpráv daného uživatele

**update** Zdroj pro zaslání nového statusu

Například:

- http://twitter.com/statuses/public_timeline.rss - zdroj obsahující veřejný seznam ve formátu RSS.
- http://twitter.com/statuses/update.xml - zdroje umožňující zaslání stavové zprávy. Odpověd serveru bude ve formátu XML.

Modul `TwitterPost` používá zdroj update. Tento zdroj má dva parametry: povinný parametr `status`, který přímo obsahuje text zprávy a nepovinný parametr `in_reply_to_status_id`, který je použit pouze v případě, že zpráva je odpovědí na již existující zprávu ať už vlastní, nebo jiného uživatele. V případě nové zprávy nemá opodstatnění a modul ho nijak nevyužívá. U parametru `status` je doporučeno omezení na délku 140 znaků, u případné delší zprávy není zaručeno její korektní zobrazení ať už přímo na webu Twitter.com nebo v dostupných klientských aplikacích. Textový obsah tohoto parametru musí být zakódován podle RFC3986[1] (tzv. URI encoding). V tomto kódování jsou povolena malá a velká písmena anglické abecedy (”A” až ”Z”, ”a” až ”z”) a číslice (”0” až ”9”), dále pak znak spojovník (”-”), podtržení (”_”), tečka (”.”) a tilda (”~”). Speciální znaky, jako například znak středník (”;”), dvojtečka (”:”), vykřičník (”!”), lomítko (”/”) a některé další mají zvláštní význam. Ostatní znaky a speciální znaky použité v jiném významu než v daném zvláštním významu musí být převedeny na procentový kód ve tvaru %HH, kde písmeno H označuje jednu číslici v šestnáctkové soustavě. HH je pak tedy osmibitové číslo, které odpovídá kódu znaku v dané národní kódovací tabulce (ASCII, ISO 8859-2, UTF-8 atd.) Twitter API zde navíc předpokládá, že všechny znaky, které nejsou obsaženy v povolených znacích byly před zakódováním podle RFC3986 kódovány v mezinárodním kódování UTF-8 a podle toho s nimi zachází. Limit 140 znaků je aplikován na nezakódovanou zprávu, přesáhne-li zakódovaná délka toto omezení, nijak se to na výsledku neprojeví.

### 4.1.3 Implementace

Modul zajišťující komunikaci se sociální sítí Twitter je realizována třídou `TwitterPost`. Jejím cílem je publikovat upozornění na nový záznam v sekci ”Aktuálně” školního informačního systému v podobě stavové zprávy v síti Twitter. Pro tyto účely byla na serveru Twitter.com založena entita s uživatelským jménem VOSIS, pod kterou budou upozornění publikována. Třída obsahuje atributy `text` a `url`, konstruktor, metodu `postStatus` odesílající zprávu a metodu `printAttr` poskytující ladící výpis.

Konstruktor Konstruktor očekává čtyři parametry: jméno autora příspěvku ve tvaru ”Příjmení Jméno”, nadpis, text příspěvku a odkaz na němž lze daný příspěvek zobrazit. Z těchto čtyř parametrů se provede naplnění atributů třídy.

Atribut text Z důvodu omezení délky stavové zprávy na 140 znaků je atribut text omezen na maximálně 110 znaků. Nejdříve se zkrátí křestní jméno autora příspěvku na iniciálu, tedy do tvaru Příjmení J. a spojí se s nadpisem a textem příspěvku. Jednotlivé části jsou odděleny mezerami a spojovníkem. Poté se z takto spojeného textu odstraní diakritika a ořízne se na 110 znaků. Toto oříznutí provádí funkce smartCut (není součástí třídy `TwitterPost`, ale obecné knihovny funkcí, která se nachází v samostatném souboru, aby bylo možno ji využít i v jiných modulech), která nejdříve

provede prosté omezení délky na 106 znaků, následně odstraní případné nekompletní slovo na konci textu a doplní na konec textového řetězce tři tečky, naznačující pokračování příspěvku.

Pokud by tedy zpráva v sekci aktuality vypadala například takto:

```
Nováková Kateřina: Exkurze pro studenty VŠE
Dne 21.3.2009 proběhne exkurze přihlášených studentů třetího ročníku
do rakouského státního archivu ve Vídni. Odjezd v 8:00 od školy.
```

Pak atribut text bude naplněn textem

```
Novakova K. - Exkurze pro studenty VSE - Dne 21.3.2009 probehne
exkurze prihlasenych studentu tretiho...
```

Atribut url Pro atribut url zbývá ve stavové zprávě 30 znaků. Většina odkazů v rámci školního webu je ale delší a nesplňovaly by tedy toto omezení. Standardní praxí je využití služeb některého ze serverů, které nabízejí zkrácení URL. Třída `TwitterPost` využívá služeb serveru TinyURL.com, který nabízí zkrácení libovolného www odkazu do tvaru http://tinyurl.com/nahodny_kod, tedy například odkaz http://www.sks.cz/oskole/aktuality.phtml je zkrácen na http://tinyurl.com/dzjhcq.

Tento server má navíc velmi jednoduché API, stejně jako v případě Twitter API je i toto rozhraní založené na HTTP protokolu. Pro získání zkráceného odkazu je potřeba odeslat na adresu zdroje http://tinyurl.com/api-create.php metodou GET parametr url, do nějž se naplní původní URL. Odpovědí serveru je pak zkrácený odkaz ve formě neformátovaného ASCII textu.

Pokud považujeme prvních 19 znaků takto zkráceného URL za fixní, pak pro náhodný kód zbývá 11 znaků. Tento kód je tvořen z alfanumerických znaků a v podstatě jeho délka narůstá pouze s celkovým počtem všech takto zkrácených URL. V současné době mají kódy obvykle šest znaků a to po více než šesti letech od založení služby. Je tedy bezpečné předpokládat, že jedenáct znaků bude v dohledné době postačovat. Interakci se službou TinyURL zajišťuje funkce getTinyURL (opět není součástí třídy samotné, ale byla umístěna souboru obsahujícího znovupoužitelné funkce).

Metoda postStatus Ke komunikaci se serverem pomocí Twitter API je použit PHP modul cURL, který umožňuje přímou komunikaci skriptu běžícího na jednom webovém serveru s jiným webovým serverem protokolem HTTP. Umí odesílat a přijímat data požadovanými metodami i ověřit uživatele metodou HTTP Basic Authentication. Metoda OAuth není implementována, nicméně bylo by ji možné v případě potřeby

doplnit na úrovni PHP skriptu. Metoda postStatus odesílá pomocí modulu cURL stavovou zprávu, kterou před odesláním převede do kódování podle RFC3986. Zpráva zaslaná zpět serverem Twitter.com není nijak dál zpracovávána, přihlíží se pouze k stavovému HTTP kódu. Pokud server vrátí kód 200 OK, pak metoda vrací hodnotu true, v případě jiného kódu je návratová hodnota metody false.

Metoda printAttr Tato metoda slouží k případnému odstraňování problému vzniklých při používání třídy TwitterPost. Vypisuje obsah atributů třídy a délku výsledné stavové zprávy. Formátování výpisu je provedeno pomocí základních tagů jazyka HTML.

## 4.2 Sociální síť Facebook

### 4.2.1 Vznik a historie sítě Facebook

Facebook je největší služba sociální sítě na světě. V současné době má téměř 200 milionů registrovaných uživatelů. Vznikl v roce 2003 na americké Harvard University jako univerzitní projekt Marka Zuckerberga, který měl za cíl sdružovat studenty této školy. Poté, co se k původnímu autorovi přidalo několik jeho kolegů z univerzity vznikl projekt, který záhy používala téměř polovina studentů Harvardu. Během několika málo měsíců se rozšířilo jeho používání i na další vysoké školy v USA a Kanadě. V roce 2005 byl umožněn přístup i studentům středních škol a zaměstnancům několika vybraných společností. V druhé polovině roku 2006 se pak stala síť Facebook přístupná pro kohokoliv na celém světě (registrace vyžaduje platnou emailovou adresu a věk od třinácti let výše) a od té doby se z ní stal jeden z největších fenoménů současného Internetu.


Obrázek 4.2: Ukázka uživatelského profilu.

Facebook umožňuje registrovaným uživatelům vytvořit svůj profil na kterém o sobě sdělí fakta, která si sami zvolí a pomocí integrovaného vyhledávače si mohou najít profily lidí, které znají a vytvořit si seznam přátel. S přáteli pak mohou sdílet nejrůznější obsah:

textové stavové zprávy, fotografie, videonahrávky, odkazy a další. Přátelům lze zároveň nabízet kalendářové události a tak je zvát na společné události. Uživatelé se mohou sdružovat do zájmových skupin například podle města v němž žijí, podle společnosti, ve které pracují, podle svých koníčků a zájmů atd. Jednou z dalších funkcí, kterou je Facebook vybaven je i online textová komunikace založená na stejných principech jako běžné tzv. instant messaging služby (Jabber, ICQ, a další). Počet těchto služeb se navíc neustále zvětšuje. Krom funkčnosti, kterou poskytuje přímo jádro této sítě, je zde ještě možnost využívání celé sady modulů, které jsou tvořeny jak přímo týmem autorů Facebooku, tak různými dalšími vývojáři. Tyto moduly umožňují interakci s dalšími weby a sociálními sítěmi.

### 4.2.2 Propojení sítí Twitter a Facebook

Z vlastní zkušenosti vím, že značná část spolužáků ze stejného ročníku již má svůj profil na Facebooku a služby této sítě ve větší či menší míře využívají. Studenti VOŠIS mají vlastní zájmové skupiny týkající se studia na VOŠIS a proto je žádoucí aby zejména aktuální dění ve škole mohli sledovat i v rámci této sítě. Facebook stejně jako síť Twitter nabízí veřejně přístupné API, pomocí kterého lze interagovat s dostupnými službami. Toto API je poměrně komplexní, neboť pokrývá mnohem více funkcí, než API sítě Twitter. Nicméně pro zveřejňování aktualit na profilu instituce, který lze pro VOŠIS vytvořit není třeba tohoto API programově využívat. Existuje totiž modul, který umí v plném znění přejímat zprávy ze sítě Twitter a zveřejňovat je. Výsledkem tedy bude okamžité zobrazení nové zprávy na profilu VOŠIS a zároveň všem, kdo budou mít tento profil přidaný ve svém seznamu přátel. Tento postup byl zvolen záměrně, neboť možnost opačného propojení, tedy zasílání zpráv sítě Facebook do sítě Twitter není možné. Zároveň platí, že síť Twitter má striktnější limity na délku i formální obsah zpráv, takže zprávy z ní budou v síti Facebook zobrazovány korektně.

# Kapitola 5

# Kontaktní informace

Školní webové stránky neposkytují jen informace o aktuálním dění, ale také i kontaktní informace na vyučující a administrativní pracovníky školy. Tyto informace by kromě webové stránky mohly být prezentovány ve standardním formátu vCard, který podporuje naprostá většina aplikací pro správu kontaktů (systémový Adresář obsažený ve Windows XP, Apple AddressBook, linuxový manažer kontaktů Kontact a další), e-mailových klientů (Microsoft Outlook, atd.) a značná část mobilních zařízení typu PDA a mobilních telefonů. Nebylo by tak nutné kontakty přepisovat, pouze stáhnout patřičný soubor a ten importovat do příslušné aplikace, respektive mobilního zařízení.

## 5.1 Formát vCard

### 5.1.1 Vznik a historie formátu vCard

Za vznikem formátu vCard byla myšlenka vytvoření jednotného standardu, který by byl elektronickou alternativou papírové vizitky. V roce 1995 vzniklo Versit Consortium, skupina společností podnikajících v oblasti informačních technologií (Apple, AT&T, IBM a Siemens). Tato skupina si dala za cíl vznik otevřeného formátu, který by umožňoval výměnu osobních údajů mezi širokou škálou zařízení a aplikací. Koncem roku 1996 byla činnost skupiny ukončena a výsledky její práce byly předány asociaci Internet Mail Consortium, která navrhuje standardy související s e-mailovou komunikací.

Počátkem roku 1997 byla publikována specifikace vCard 2.1[23]. V následujícím roce byly pod hlavičkou IETF vydány specifikace RFC2425[11] a RFC2426[4], které popisují vCard 3.0 jako internetový standard. Většina software a zařízení tento formát podporuje, a tak dodnes zůstává nejrozšířenějším používaným formátem pro elektronickou výměnu osobních údajů.Pro implementaci byl vybrán formát vCard 2.1, který je sice starší avšak oproti vCard 3.0 má několik výhod:

- Je podporovaný větším množstvím aplikací a zařízení. Běžně používané aplikace sice zvládají importovat oba formáty, ale pro export je většinou zvolen právě verze 2.1.
- Každé položce je možné jednoznačně určit použité národní kódování. Ve formátu vCard 3.0 bylo kódování přesunuto do hlavičky MIME typu a tak u uloženého souboru není možné zpětně zjistit použité kódování. V RFC2426 je sice uvedeno, že autoři normy preferují UTF-8, avšak toto kódování není stanoveno jako výchozí a záleží tak čistě na klientské aplikaci, jak znaky mimo rozsah sedmibitové kódovací tabulky ASCII při zpracování vizitky interpretuje.

### 5.1.2 Popis formátu

Jedná se o jednoduchý čistě textový formát. Případné binární části musí být zakódovány do textové podoby kódováním Base64. Každá elektronická vizitka začíná položkou BEGIN:VCARD, dále následuje řádek označující použitou verzi formátu, tj. v případě vCard 2.1 je to VERSION:2.1. Za těmito úvodními řádky jsou samotná data týkající se kontaktu. Vizitka je zakončena položkou END:VCARD.

Data jsou tvořena položkami ve tvaru KLÍČ;PARAMETRY:HODNOTA, přičemž každý řádek souboru obsahuje právě jednu takovou položku. Pokud je parametrů více, jsou odděleny středníkem. Totéž platí pro oddělování částí hodnoty, pokud se skládá z více částí.

Formát počítá se základními popisnými daty jako je jméno, příjmení, funkce, adresa, jméno a adresa zaměstnavatele, množství typů telefonních a faxových datových čísel (pracovní, soukromé atd.), ale také umožňuje přiřadit osobě i fotografii, logo, geografická data nebo zvukový záznam (lze využít například k objasnění výslovnosti jména).

Dále existují rozšíření, která nejsou přímo součástí standardu, ale zavedli je výrobci software a zařízení. Všechny položky, které nejsou obsaženy ve specifikaci vCard musí začínat prefixem "X-". Můžeme mezi nimi najít například jméno manžela/manželky nebo kontaktní údaje pro internetové komunikační nástroje typu instant messaging (Jabber, AIM, ICQ, MSN). Toto relativně velké množství použitelných údajů je v následujícím popisu omezeno pouze na ty typy hodnot, které byly použity v implementaci. Ostatní údaje školní databáze neudržuje a proto nebyly použity:

| | |
|---|---|
| FN | Celé jméno kontaktu včetně titulů ve tvaru určeném pro zobrazení. |
| N | Oddělené části jména, určené pro řazení v klientském programu. |
| TEL | Telefonní číslo. Specifikace nepředepisuje žádný konkrétní formát telefonního čísla. Praxí je formátovat číslo tak, jak je běžné v dané oblasti. |
| **EMAIL** | E-mailová adresa. |
| **URL** | Adresa webových stránek kontaktu. |

### 5.1.3 Implementace

Elektronickou vizitku ve formátu vCard implementuje třída `VisitCard`. Tato třída obsahuje atributy `card` a `filename`, `konstruktor`, a metody `getFileName` a `getCard`.

**Konstruktor** Očekává dva parametry: Prvním je číselná proměnná `id`, která odpovídá identifikačnímu číslu uživatele v databázové tabulce prava, evidující uživatele informačního systému, s oprávněním do něj přispívat (tj. vyučující, studijní referentky, vedení školy atd.). Pokud je tento parametr zadán, vybere z databáze kontaktní údaje pro odpovídající osobu a na základě nich sestaví vizitku, kterou uloží do atributu `card`. Není-li parametr vyplněn, pak se z tabulky automaticky vyberou všechny osoby, které mají vyplněn alespoň jeden údaj z trojice: telefonní číslo, e-mailová

adresa, adresa webových stránek a sestaví se vizitka obsahující v jednom souboru všechny tyto osoby. Více kontaktů v jednom souboru nepodporují všechny aplikace, které se soubory vCard umí pracovat, avšak jejich značná část nemá s hromadným zpracováním obtíže a proto byla tato možnost implementována. Výsledná hromadná vizitka se také uloží do atributu `card`.

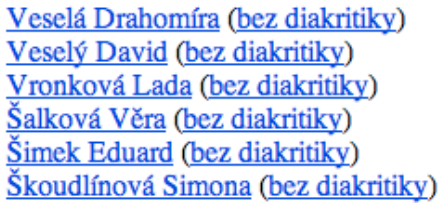
Je-li některý ze zmíněné trojice údajů nevyplněn, pak se jak u jednotlivé, tak u hromadné vizitky položka, respektive řádek s touto položkou do výsledného souboru negenerují. V případě hromadné vizitky může tento postup ušetřit objem přenesených dat. Konstruktor také rozhoduje o pojmenování výsledného souboru. Pokud se jedná o jednotlivou vizitku, pak má soubor jméno shodné s příjmením osoby v něm obsažené, jediný rozdíl je v odstranění diakritiky z názvu souboru. Jde-li o vizitku hromadnou, jmenuje se soubor kontakty_vosis. Přípona souboru je v obou případech vcf, což je standardní přípona pro soubory vizitek.

Druhým parametrem, který je třeba konstruktoru předat je parametr `ascii`, jenž je typu boolean. Je-li tento parametr nastaven, výstupem bude vizitka bez udaného kódování a zbavená diakritiky. V opačném případě je vizitka kódovaná národním kódováním Windows-1250 a to je také v příslušných položkách označeno.

**Metoda getFileName** Vrací hodnotu atributu `filename`.

**Metoda getCard** vrací hodnotu atributu `card`.

Byl vytvořen skript, který vytvoří abecedně řazený jmenný seznam všech uživatelů, kteří mají vyplněnu alespoň jednu kontaktní informaci. Tento seznam je vygenerován v podobě HTML stránky s odkazy na vizitky jednotlivých uživatelů a to jak ve formě s diakritikou, tak ve formě bez ní. Seznam zároveň obsahuje možnost vygenerovat vizitku se všemi kontakty najednou.




Obrázek 5.1: Ukázka seznamu kontaktů generovaného skriptem.

Samotný skript, který generuje vizitku musí řešit ještě jeden problém a tím je tvorba správných MIME hlaviček. Skript nastaví hlavičku označující korektně MIME typ pro vCard 2.1, který je text/x-vcard. Dále se ještě použije hlavička Content-Disposition, která při naplnění hodnotou attachment přiměje cílový webový prohlížeč aby vizitku neotevíral (jelikož jde o textový soubor, mohlo by se tak v některých prohlížečích dít), ale uložil ji. Tato hlavička zároveň změní jméno souboru z jména skriptu na jméno vrácené metodou `getFileName` třídy `VisitCard`.

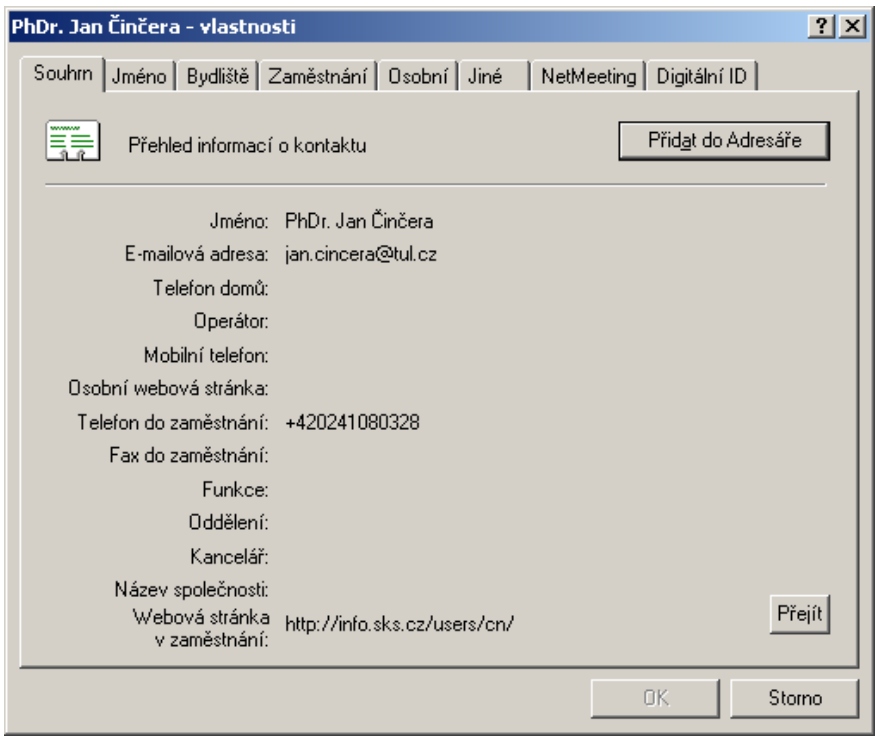
### 5.1.4 Ukázka vCard

Tento kód byl vytvořen výše popisovanými skripty a jde o elektronickou vizitku ve formátu vCard 2.1.

```
BEGIN:VCARD
VERSION:2.1
FN;CHARSET=Windows-1250:PhDr. Jan Činčera
N;CHARSET=Windows-1250:Činčera;Jan
TEL;WORK:+420241080328
EMAIL;WORK;INTERNET:jan.cincera@tul.cz
URL;WORK:http://info.sks.cz/users/cn/
END:VCARD
```

### 5.1.5 Příklad použití

Kontaktní informace v prostředí Microsoft Windows je možné zpracovávat v integrované aplikaci Adresář. Tato aplikace umí formát vCard zpracovávat oběma směry, tj. data z něj importovat i je do něj exportovat.




Obrázek 5.2: Kontaktní informace v aplikaci Adresář.

# Kapitola 6

# Kalendářové informace

Školní informační systém poskytuje řadu informací o událostech, které ve škole probíhají. Zejména se jedná o informace o naplánovaných zápočtových testech a zkouškách a dále pak také o výuce pro dálkově studující studenty. Tyto informace by bylo možné prezentovat i formou standardního kalendářového formátu.

## 6.1 Formát iCalendar

### 6.1.1 Vznik a historie formátu

Formát iCalendar vznikl pod původním názvem vCalendar za velmi podobných okolností jako formát elektronických vizitek vCard, uvedený v předchozí kapitole. I tento formát je produktem skupiny společností, které pod názvem Versit Consortium v roce 1996 přijaly za cíl vytvoření sady formátů pro snadnou elektronickou výměnu tzv. PIM (Personal Information Management) údajů. Část PIM informací, které se dotýkají plánovaní času pokrýval právě formát vCalendar.

Poté, co výsledky práce Versit Consortia přešly pod Internet Mail Consortium byl v roce 1997 publikován formát vCalendar 1.0. Koncem roku 1998 byl název oficiálně změněn na iCalendar (interně je označován jako VCALENDAR 2.0) a internetová standardizační organizace IETF pro jeho rozvoj vytvořila pracovní skupinu Calendaring and Scheduling Working Group. Ta téhož roku publikovala sadu dokumentů popisujících práci s tímto formátem - RFC2445[5], které popisuje samotný formát a RFC2446 a RFC2447, které popisují protokoly, kterými má být tento formát vyměňován mezi zařízeními a aplikacemi.

### 6.1.2 Popis formátu

Jedná se o formát v podobě čistého textu (specifikace počítá se sedmibitovým ASCII nebo kompatibilními kódováními), který pokrývá řadu kalendářových událostí (komponent):

**Událost (VEVENT)** Běžný kalendářový záznam, který má svůj název, začátek, konec a další parametry.

**Úkol (VTODO)** Umožňuje definovat úkol. Úkolu lze nastavit prioritu, časovou hranici do které musí být splněn, lze nastavit příznak označující, zda se úkol týká přímo osoby, která jej má zadaný v kalendáři, či zda je určen pro někoho jiného atd.

**Deník (VJOURNAL)** Umožňuje přiřadit určitému datu text, který popíše události daného (proběhlého) dne.

**Volný čas / Obsazený čas (VFREEBUSY)** Ačkoliv mají všechny kalendářové události výchozí stav, který oznamuje zda během jejich průběhu je osoba pro další případné aktivity zaneprázdněna či nikoliv, je možné událostem explicitně tento stav nastavit.

**Upomínka (VALARM)** Pokud má některá událost vyvolat vizuální, zvukové či jiné upozornění ze strany klientské aplikace či zařízení, pak je třeba použít tento typ kalendářové události.

**Časové pásmo (VTIMEZONE)** Nastavení odpovídajícího časového pásma.

Výše uvedené kalendářové komponenty mají množství parametrů, které specifikují vlastnosti konkrétních událostí. Události, které školní informační systém poskytuje, je možné pokrýt pomocí komponenty VEVENT. Všechny parametry této komponenty jsou nepovinné, v následujícím výčtu jsou uvedeny jsou pouze ty, které byly použity při implementaci:

**DTSTART** Datum začátku události. Formát data je RRRRMMDDTHHMMSS, kde RRRR je rok, MM je měsíc, DD je den, T je oddělovač data od času, HH je hodina, MM je minuta a SS je sekunda času začátku konání události

**DTEND** Datum konce události. Datum je stejné jako u parametru DTSTART.

**SUMMARY** Stručný textový popis události. Standard nespecifikuje, co má obsahovat, ovšem většina aplikací pracujících s kalendářovými daty jej používá jako název události, proto by neměl být příliš dlouhý. V případě potřeby delšího popisu je možno využít parametr DESCRIPTION, který může obsahovat plný popis události.

**LOCATION** Místo, kde se událost koná. Může obsahovat plnou adresu objektu, odkaz na vizitku, která tuto adresu obsahuje, případně může obsahovat jen konkrétní specifikaci (např. místnost), pokud se předpokládá, že je adresa známá.

Tyto parametry smí být v rámci každé události uvedeny pouze jednou. Komponenta VEVENT má ještě celou řadu dalších parametrů, například RELATED-TO, který označuje které dvě události se k sobě vztahují. V jednoduchém seznamu událostí, jakým jsou seznam zkoušek a rozvrh dálkového studia ovšem nemají hlubší význam a některé z nich ani pomocí dostupných dat naplnit nelze.

Každý řádek kalendářového souboru obsahuje jednu položku ve tvaru KLÍČ:HODNOTA. Soubor musí začínat povinnou hlavičkou, kterou tvoří začátek kalendáře ve tvaru BEGIN:VCALENDAR, dále následuje použitá verze formátu, v případě formátu iCalendar je to VERSION:2.0 (jak bylo uvedeno v předchozím textu, formát iCalendar je chápán jako verze 2.0 formátu vCalendar). Za označením verze musí následovat označení aplikace, která kalendář vygenerovala. Toto označení by mělo být unikátní pro každou aplikaci, z příkladů v RFC2445 se lze dovodit jednoduchého schématu tvorby označení, které obsahuje autora, název, jazyk a případně verzi aplikace: PRODID:-//VOSIS//iCalendarGenerator//CZ. Za těmito úvodními informacemi o kalendáři následují kalendářové komponenty. Komponenty povinně začínají uvozením typu

komponenty, v případě události VEVENT je to BEGIN:VEVENT. Následují parametry události, po nichž je komponenta ukončena položkou END:VEVENT. Událostí typu VEVENT může být v kalendářovém souboru vyšší množství, standard jejich počet nijak neupravuje ani neomezuje. Soubor je ukončen zakončením kalendáře ve tvaru END:VCALENDAR.

### 6.1.3 Implementace

Kalendář ve formátu iCalendar je implementován třídou `Calendar`. Tato třída má atributy `ascii`, `calendar`, `count`, `date` a `table`, konstruktor a metody `countEvents`, `loadEvents`, `getCount`, `doHeaders` a `getCalendar`.

**Konstruktor** očekává parametry `caltype` a `ascii`. Podle hodnoty `caltype` je vybrána zdrojová tabulka. V případě, že je tento parametr nastaven na hodnotu "dalkove" vybere se jako zdroj kalendářových událostí tabulka obsahující rozvrh studia pro dálkově studující studenty. Pokud parametr nabývá jakékoliv jiné hodnoty nebo není vůbec naplněn je zvolena tabulka obsahující termíny zkoušek. Název tabulky je uložen do atributu `table`. Parametr `ascii` určuje, zda se mají data být ve formátu sedmibitového kódování ASCII, nebo v mezinárodním kódování UTF-8. Specifikace formátu udává, že obě kódování jsou možná (stejně jako některá další kódování, která jsou s ASCII kompatibilní), avšak pro případ, že by některý klient neuměl zpracovat data v kódování UTF-8 je výhodné tuto možnost obsáhnout. Hodnota tohoto parametru je naplněna do atributu `ascii`. Konstruktor zároveň do atributu date uloží aktuální datum v okamžiku vzniku instance třídy, pro omezení kalendářových událostí ve všech použitých dotazech. Všechny dotazy pracují pouze s událostmi, jejichž datum je větší nebo rovno datu vzniku instance. V potaz se bere pouze datum, údaj o čase je ignorován.

**Metoda countEvents** nemá žádný parametr. Zjišťuje počet kalendářových událostí ve vybraném zdroji a tento počet ukládá do atributu `count`. Byla do definice třídy doplněna z informativních důvodů. Samotný kalendář (na rozdíl od vizitek) totiž není stahován klientskou aplikací na disk daného uživatele, ale v ideálním případě je odkaz na něj přímo zadán aplikaci a ta si data sama stahuje přímo ze serveru. Proto je vhodné nejdříve vytvořit seznam kalendářů v podobě webové stránky a do něj mimo jiné uvést například i počet událostí v daném kalendáři.

**Metoda loadEvents** opět nevyžaduje žádné parametry (platí to i pro všechny následující metody) a jejím účelem je naplnit událostmi z vybraného zdroje atribut `calendar`. Naplnění parametrů kalendářové komponenty VEVENT probíhá specificky, protože data v databázové tabulce nejsou uložena tak, aby jeden záznam odpovídal jedné události. Jednotlivé záznamy v tabulce odpovídají půlhodinovým blokům, ze kterých se může ve školním informačním systému událost skládat, přičemž

každý záznam má uložen začátek takového půlhodinového bloku. Metoda loadEvents tedy nejdříve vyhledá jednotlivé události (nezávisle na tom, z kolika půlhodinových bloků se skládají) a pro každou událost vyhledá její první a poslední půlhodinový blok. Z nich určí začátek a konec celé události a těmi naplní hodnoty polí DTSTART a DTEND. Do pole SUMMARY je uložen text skládající se z kódu předmětu, typu události (zkouška atd.), jména nebo zkratky jména vyučujícího (dozoru při zkoušce), kódu skupiny studentů, které se daná událost týká (D pro dálkové studium, B pro bakalářské studium atd.), a poznámky, kterou vyučující k záznamu připojil. Pole LOCATION je naplněno označením místnosti, ve které se událost bude odehrávat.

**Metoda getCount** vrací hodnotu atributu `count` nastaveného metodou `countEvents`. Do struktury třídy byla doplněna z důvodu přehlednosti, všechny metody které mají návratovou hodnotu začínají prefixem get.

**Metoda getCalendar** vrací obsah atributu `calendar`.

**Metoda doHeaders** předává klientské aplikaci korektní HTTP hlavičku. V této hlavičce je nastaven MIME typ dokumentu podle specifikace na text/vcalendar a podle atributu ascii je přidáno označení použitého kódování, tj. UTF-8 v případě použití diakritiky (atribut `ascii` není nastaven) nebo bez označení kódování v případě, že diakritika není použita.

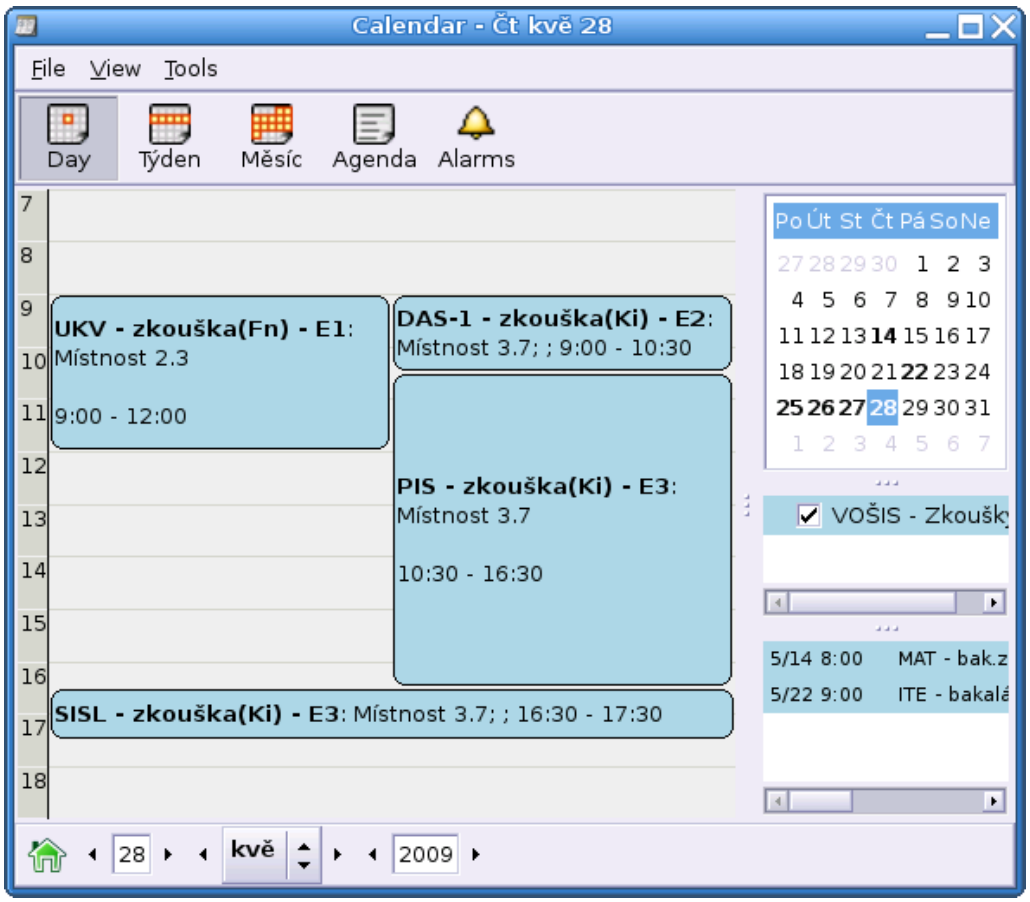
### 6.1.4 Ukázka iCalendar

Následující kód byl vytvořen výše popisovanými skripty a jde o elektronický kalendář ve formátu iCalendar (vCalendar 2.0).

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//VOSIS//iCalendarGenerator//CZ
BEGIN:VEVENT
DTSTART:20090514T080000
DTEND:20090514T093000
SUMMARY:MAT - bak.zkouška, MAT-2 - B - zápočet(Pa) - B
LOCATION:Místnost 2.3
END:VEVENT
EVENT:VCALENDAR
```

### 6.1.5 Příklad použití

V prostředí operačního systému Linux lze online kalendáře zpracovávat například aplikací GPE Calendar (v českých variantách OS Linux je název přeložen jako Kalendář). Aplikace umí odebírat více online kalendářů, data z nich zobrazovat po dnech, týdnech, měsících nebo jako přehled agendy z kalendářů vyplývající.




Obrázek 6.1: Online kalendář zkoušek v aplikaci GPE Calendar.

## 6.2 Google Calendar

### 6.2.1 Popis služby Google Calendar

Google Calendar je online plánovací kalendář, který umožňuje pomocí webového rozhraní komplexní organizaci času. Je v něm možné vytvořit si sadu několika vlastních kalendářů a v nich evidovat kalendářové události. Je tak možné oddělit soukromé, pracovní, školní a další kalendáře. Navíc lze kalendáře mezi jednotlivými uživateli sdílet, případně si do soukromého seznamu kalendářů zařadit některý z kalendářů veřejných.

Díky těmto vlastnostem a některým dalším přidaným službám, jako je například upozorňování na nastávající události pomocí krátkých textových zpráv na zadaný mobilní telefon zdarma je tato služba velmi populární. Rostoucí popularitu způsobuje u fakt, že

se jedná o plně webovou aplikaci, kterou je možné používat odkudkoliv, kde je zajištěn přístup na Internet. Data v ní vytvořená lze exportovat a používat i v offline kalendářovém software.

### 6.2.2 Google Calendar API a Zend Framework

Aplikační rozhraní Google Calendar využívá stejně jako rozhraní všech ostatních veřejně přístupných služeb, které Google poskytuje (například Blogger, Google Maps atd.), protokol GData. Tento protokol je založen syndikačním formátu a publikačním protokolu Atom s několika rozšířeními. V praxi není třeba pracovat přímo s tímto protokolem, neboť většina programovacích jazyků, které se v prostředí tvorby dynamických webových stránek používají má knihovny, které interakci se servery Google pomocí GData protokolu umožňují. Pro jazyk PHP zprostředkovává tuto funkčnost Zend Framework, který obsahuje třídy, pomocí nichž lze interagovat i s kalendářovými službami.

### 6.2.3 Implementace

Používání výše uvedeného frameworku je komplikované a způsobuje neúměrný nárůst délky kódu. Každá manipulace s kalendářovými daty znamená naplnění datových struktur patřičného objektu a posléze zavolání příslušné metody. Výsledkem implementace Google Calendar API je proto knihovna `CalFunctions`, která obsahuje jednotlivé funkce, které mohou být použity pro práci s Google Calendar v rámci školního IS. V rámci implementace nebyla služba Google Calendar propojena s žádnou fungující částí informačního systému, vzhledem k množství různých událostí, které by bylo možné do tohoto online kalendáře ukládat je takovéto propojení věcí další úvahy.

**calLogon** Tato funkce zajišťuje zabezpečené přihlášení ke službě Google Calendar. Má dva parametry - `user` a `pass` - do kterých je potřeba naplnit uživatelské jméno a heslo. Vrací objekt, který reprezentuje vytvořené ověřené spojení. Tento objekt je potřeba předat prakticky při jakékoliv činnosti, neboť bez něj server považuje přístup za neoprávněný. Výjimkou jsou pouze čtecí operace na veřejných kalendářích, kde není ověření třeba.

**calCreateEvent** Vytvoření nové kalendářové položky zajišťuje tato funkce. Má parametry `client`, který odkazuje na vytvořené spojení, dále `visibility` označující zda je nový záznam veřejný nebo soukromý, `title` obsahující název události a `desc`, v němž je popis události. Dvojice parametrů `start` a `end` určuje dobu konání události a `where` popisuje místo konání.

**calDeleteEventSingle** Pomocí této funkce lze smazat jednotlivý záznam. Tento záznam je specifikován druhým parametrem `eventId`, prvním parametrem je `client`, který představuje ověřené spojení se serverem.

**calDeleteEventMultiple** Pokud je třeba smazat více záznamů najednou, lze tak učinit zavoláním této funkce. Kromě ověřeného spojení má funkce ještě parametr `eventFeed`, což je seznam událostí, které se mají z kalendáře odstranit.

**calEventsAll** Tato funkce vrací pole obsahující všechny budoucí kalendářové položky, které obsahuje výchozí kalendář uživatele přihlášeného pomocí funkce `calLogon`. Tyto položky jsou u této stejně jako všech dalších vyhledávacích funkcí řazeny vzestupně podle času označujícího začátek události. Má dva parametry - `client`, do nějž je třeba naplnit objekt reprezentující ověřené spojení a dále parametr `visibility`, kterým lze filtrovat zda se mají vybrat veřejné nebo soukromé položky z kalendáře. Lze nastavit na hodnoty "public" nebo "private".

**calEventsByDate** Umožňuje filtrovat kalendářové záznamy podle data začátku. Má čtyři parametry, první dva - `client` a `visibility` - mají stejný význam jako u funkce `calEventsAll`, další dva `datemin` a `datemax` pak spodní a horní mez časového intervalu, do kterého musí začátek události patřit. Vrací pole obsahující položky, které zadaným kritériím vyhovují.

**calEventsFullText** Umožňuje plnotextově prohledávat kalendářové položky. Má tři parametry, první mají opět stejný význam jako u funkce `calEventsAll`, třetím parametrem je `ftQuery`, do nějž se plní text, který chceme v kalendářových událostech vyhledat. Funkce vrací pole, v němž jsou položky, které zadaný text obsahují.

**calUpdateEvent** Pokud je třeba změnit některý z parametrů, je možné k tomu použít tuto funkci. Parametry jsou shodné jako u funkce `calCreateEvent`, mezi parametry `visibilty` a `title` je vložen parametr `eventId`, který odkazuje na příspěvek, jehož parametry mají být upraveny.

Použití těchto funkcí je demonstrováno ve skriptu `index.php`, který vytváří v kalendáři testovací položky, pracuje s nimi a posléze je smaže.

# Kapitola 7

# Závěr

V průběhu této práce se podařilo pochopit strukturu databáze a školního informačního systému vůbec. Výsledkem je funkční implementace syndikačních formátů Atom a RSS, elektronických vizitek vCard a kalendáře ve formátu iCalendar. Skripty generující tyto formáty jsou přímo napojeny na skutečná data obsažená v školním informačním systému a je možné je podle úvahy nasadit do praxe. Interakce se službami sociální sítě sítí Twitter a zprostředkovaně i se sociální sítí Facebook je dopracována do stádia použitelné třídy, kterou je možné zapracovat do stávajícího systému. Pro interakci s webovou kalendářovou aplikací Google Calendar byla zpracována knihovna funkcí, které je možné v budoucnosti použít.

Při řešení úkolů souvisejících s touto prací jsem narazil i na některé problémy:

- Testovací server work, na kterém bylo možné pracovat s programovacím jazykem PHP 5 neumožňuje spolupracovat s ODBC ovladači, pomocí kterých lze přistupovat k databázovým souborům FoxPro. Databáze FoxPro uchovává záznamy o rozvrhu a zapsaných studentech. Proto se implementace kalendářového formátu iCalendar týká jen těch událostí, které jsou uloženy v MySQL databázi, ačkoliv původně bylo počítáno se všemi kalendářovými událostmi. V souvislosti s kalendářovými událostmi proto vyvstala otázka, zda by nemohly být záznamy v budoucnosti duplikovány i do MySQL databáze, aby se webová prezentace oddělila od souborů FoxPro. Samotnou databázi FoxPro není možné eliminovat, neboť je na ni napojen software generující rozvrhy a eliminující kolize mezi kurzy. Protože se bude jednat o zdvojování dat, je možné uvážit i částečnou změnu struktury informací o rozvrhu, tak aby korespondovala s tabulkami, které již v MySQL databázi jsou (tj. informace o zkouškách a dálkovém studiu).

- Všechny informace jsou v tabulkách uloženy v národním kódování Windows Latin 2 (Windows Code Page 1250). Některé formáty předpokládají nebo přímo ze specifikace vyžadují kódování jiné. Bylo proto třeba vytvořit funkce, které zbavují text kódování, nebo převádějí Windows Latin 2 na UTF-8.

- Pro export kalendářových dat se původně počítalo s formátem vCard 3.0, avšak po vytvoření skriptu jsem zpětně zjistil obtíže s kódováním. Generované vizitky se korektně importovaly pouze na platformě Microsoft Windows, v aplikacích pod jinými operačními systémy docházelo k poškození informací obsahujících české znaky. Formát vCard 3.0 neobsahuje informace o použitém kódování a tak je výběr kódování

čistě v režii příslušného software. Proto jsem skript upravil tak, aby generoval starší formát vCard 2.1, který tyto informace obsahuje.

- Testovací server work nemá nainstalovaný Zend Framework a po jeho instalaci PHP vracelo chybu, podle které nejsou korektně nastaveny zabezpečené SSL přenosy, které se používají při komunikaci se servery Google. Jelikož implementace Google Calendar byla dovedena pouze do stádia obecných funkcí, nepředstavovalo problém zkoušet kód na vlastním serveru mimo školní síť. Pokud by ovšem bylo rozhodnuto o nasazení těchto funkcí, bylo by třeba Zend Framework nainstalovat korektně a nastavit zabezpečené přenosy.

- V případě, že by školní systém měl někdy publikovat příspěvky do sítě Facebook, bylo by třeba vytvořit oficiální účet VOŠIS. Tento účet může podle podmínek používání serveru Facebook.com vytvořit pouze osoba pověřená vedením instituce.

Rámec této práce neobsáhl nasazení návrhů do praktického provozu a předání uživatelům. V případě, že se správa školního informačního systému pro toto nasazení rozhodne, je možné v budoucnu na rozšíření popsané funkcionality dále spolupracovat.

# Použitá literatura


[1] BERNERS-LEE, T. – FIELDING, R. T. – MASINTER, L. *RFC 3986 : Uniform Resource Identifier (URI): Generic Syntax (Obecná syntaxe URI)* [online]. [cit. 21.03.2009]. Popis syntaxe URI. Dostupné z: `http://tools.ietf.org/html/rfc3986`.

[2] BRÁZA, J. *PHP 4 : Praktické příklady.* 1. vydání. Praha : Grada, 2003. Konkrétní příklady použití PHP v součinnosti s XML, databázemi a dalšími webovými technologiemi. ISBN 80-247-0441-2.

[3] *Builder.cz : diskusní fórum [diskusní fórum]* [online]. [cit. 21.03.2009]. Diskusní fórum zabývající se programováním obecně i konkrétními programovacími jazyky. Dostupné z: `http://forum.builder.cz/`.

[4] DAWSON, F. – HOWES, T. *RFC 2426 : vCard MIME Directory Profile (Adresářový MIME profil vCard)* [online]. [cit. 21.03.2009]. Specifikace MIME profilu datového formátu vCard 3.0. Dostupné z: `http://tools.ietf.org/html/rfc2426`.

[5] DAWSON, F. – STENERSON, D. – GANGULY, A. *RFC 2445 : Internet Calendaring and Scheduling Core Object Specification (Specifikace formátu iCalendar)* [online]. [cit. 01.05.2009]. Popis formátu iCalendar. Dostupné z: `http://tools.ietf.org/html/rfc2445`.

[6] DUBOIS, P. *MySQL profesionálně : Kompletní průvodce použitím, programováním a správou MySQL.* 1. vydání. Praha : Mobil Media, 2003. Popis funkčnosti MySQL, aplikačního rozhraní, interakce s PHP a administrace tohoto databázového serveru. ISBN 80-86593-41-X.

[7] FIELDING, R. T. *Architectural Styles and the Design of Network-based Software Architectures (Design a styly síťových aplikací)* [online]. 2000. [cit. 21.03.2009]. Popis specifikace REST. Dostupné z: `http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm`.

[8] *Google Calendar APIs and Tools (Nástroje a API služby Google Calendar)* [online]. [cit. 21.03.2009]. Popis aplikačního rozhraní služby Google Calendar a nástrojů, které lze použít pro interakci s ním. Dostupné z: `http://code.google.com/apis/calendar/`.

[9] *Google Groups : Google Calendar Data API (Skupiny Google : Datové aplikační rozhraní Google Calendar)* [online]. Diskusní fórum zaměřené na API Google Calendar a na odstraňování problému vzniklých při jeho používání. Dostupné z: `http://groups.google.com/group/google-calendar-help-dataapi`.

[10] HORNÝ, S. *Vybrané kapitoly systémové metodologie*. 1. vydání. Praha : Oeconomica, 2003. ISBN 80-245-0599-1.

[11] HOWES, T. – SMITH, M. – DAWSON, F. *RFC 2425 : A MIME Content-Type for Directory Information (MIME typ pro adresářové informace)* [online]. [cit. 01.05.2009]. Specifikace MIME typu pro adresářové informace. Dostupné z: `http://tools.ietf.org/html/rfc2425`.

[12] *MySQL Documentation (MySQL Dokumentace)* [online]. [cit. 21.03.2009]. Dokumentace k serveru MySQL, popis jazyka a API. Příklady. Dostupné z: `http://dev.mysql.com/doc/`.

[13] *Personal Data Interchange* [online]. [cit. 21.03.2009]. Webová stránka sdružující dokumentaci k datovým formátů, určeným pro výměnu osobních dat, zejména kontaktních a kalendářových informací. Dostupné z: `http://www.imc.org/pdi/`.

[14] *PHP Manual* [online]. [cit. 21.03.2009]. Kompletní online dokumentace ke skriptovacímu jazyku PHP. Dostupné z: `http://www.php.net/manual/en/`.

[15] *RFC 4287 : The Atom Syndication Format (Syndikační formát Atom)* [online]. [cit. 21.03.2009]. Specifikace syndikačního formátu Atom určeného ke sledování změn obsahu webových stránek. Dostupné z: `http://tools.ietf.org/html/rfc4287`.

[16] *RFC 5023 : The Atom Publishing Protocol (Publikační protokol Atom)* [online]. [cit. 21.03.2009]. Specifikace publikačního protokolu Atom určeného ke sledování změn obsahu webových stránek. Dostupné z: `http://tools.ietf.org/html/rfc5023`.

[17] *RSS 2.0 Specification* [online]. [cit. 03.04.2009]. Aktulní verze spefikace standardu RSS. Dostupné z: `http://www.rssboard.org/rss-specification`.

[18] *RSS History* [online]. Vznik a historie standardu RSS. Dostupné z: `http://www.rssboard.org/rss-history`.

[19] RUBY, S. *Anatomy of a Well Formed Log Entry* [online]. 2003. [cit. 21.03.2009]. Úvaha na téma syndikačních formátů, která započala vývoj formátu Atom. Dostupné z: `http://www.intertwingly.net/blog/1472.html`.

[20] *Ing. Antonín Skopec*, zástupce ředitele pro informační systém a rozvoj. Kontakt: skopec@sks.cz. Konzultace v období březen - květen 2009.

[21] TVEIT, H. *RFC 1766 : Tags for the Identification of Languages (Jazykové identifikační značky)* [online]. [cit. 17.04.2009]. Specifikace označování jazyků a jazykových dialektů. Dostupné z: `http://tools.ietf.org/html/rfc1766`.

[22] *Twitter API wiki (Dokumentace Twitter API)* [online]. [cit. 21.03.2009]. Popis aplikačního rozhraní sociální sítě Twitter. Dostupné z: `http://apiwiki.twitter.com/`.

[23] *vCard - The Electronic Business Card - Version 2.1* [online]. 1996. [cit. 01.05.2009]. Specifikace formátu elektronické vizitky vCard 2.1. Dostupné z: `http://www.imc.org/pdi/vcard-21.txt`.

[24] *Visual FoxPro Developer Center* [online]. Vývojářské dokumentační centrum pro databázi Fox Pro a vývojařské nástroje s ní související. Dostupné z: `http://msdn.microsoft.com/en-us/vfoxpro/default.aspx`.

[25] *Vyšší odborná škola informačních služeb* [web site]. Dostupné z WWW: `http://www.sks.cz` Webové stránky školy, informace o škole, přístup do informačního systému školy.

[26] WELLING, L. – THOMSON, L. *PHP a MySQL : Rozvoj webových aplikací.* 3. vydání. Praha : Softpress, 2005. Popis základních vlastností PHP a MySQL a jejich instalace. Tvorba aplikací v PHP a MySQL včetně příkladů. Zabezpečení webových aplikací. ISBN 80-86497-83-6.

[27] WILLIAMS, H. E. – LANE, D. *PHP a MySQL : vytváříme webové databázové aplikace.* 1. vydání. Praha : ComputerPress, 2002. Tvorba aplikací v PHP a MySQL. Kniha obsahuje komplexní příklad tvorby webového informačního systému. ISBN 80-7226-760-4.

[28] *World Wide Web Consortium : Web standards* [online]. [cit. 21.03.2009]. Webové stránky WWW konzorcia, které vytváří standardy používané na webu. Dostupné z: `http://www.w3.org/`.