

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Magisterská práce

Průvodce studenta FIT ČVUT

Bc. Jan Molnár

Vedoucí práce: Ing. Michal Havryluk

6. května 2012

Poděkování

Doplňte, máte-li komu a za co děkovat. V opačném případě úplně odstráňte tento příkaz.

Prohlášení

Prohlašuji, že jsem tuto práci vytvořil samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), nemám závažný důvod proti užití tohoto školního díla a k užití uděluji svolení.

V Praze dne 6. května 2012

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2012 Jan Molnár. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Jan Molnár. *Průvodce studenta FIT ČVUT: Magisterská práce*. Praha: ČVUT v Praze, Fakulta informačních technologií, 2012.

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahradte seznamem klíčových slov v angličtině oddělených čárkou.

Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

Klíčová slova Nahradte seznamem klíčových slov v češtině oddělených čárkou.

Obsah

Úvod	17
1 Popis problému, specifikace cíle	19
1.1 Popis řešeného problému	19
1.2 Popis uživatele a jeho potřeb	20
1.3 Vymezení cílů diplomové práce	20
1.4 Popis struktury DP ve vztahu k vytyčeným cílům	21
1.5 Serverová aplikace	21
1.5.1 Rešeršní zpracování existujících ontologií	22
1.6 Mobilní aplikace	23
1.6.1 Rešeršní zpracování existujících mobilních aplikací	23
2 Analýza a návrh implementace	27
2.1 Serverová aplikace	27
2.1.1 Potřeby uživatelů	27
2.1.2 Případy užití	28
2.1.3 Funkční požadavky	32
2.1.4 Nefunkční požadavky	34
2.1.5 Návrh ontologie	35
2.1.6 Návrh architektury	35
2.1.7 Návrh řešení	37
2.1.8 Návrh bezpečnosti	37
2.1.9 Použité technologie	39
2.1.10 Použité knihovny	40
2.1.11 Použité nástroje	41
2.2 Mobilní aplikace	43
2.2.1 Potřeby uživatelů	43
2.2.2 Případy užití	45
2.2.3 Funkční požadavky	48
2.2.4 Nefunkční požadavky	49
2.2.5 Návrh architektury	49
2.2.6 Návrh řešení	49
2.2.7 Návrh bezpečnosti	49

2.2.8	Návrh uživatelského rozhraní	50
2.2.9	Použité technologie	50
2.2.10	Použité knihovny	52
2.2.11	Použité nástroje	53
3	Realizace	55
3.1	Serverová aplikace	55
3.1.1	Řešené problémy	55
3.1.2	Implementace bezpečnosti	55
3.2	Mobilní aplikace	55
3.2.1	Řešené problémy	55
3.2.2	Implementace bezpečnosti	56
3.2.3	Ideální implementace	56
4	Testování	59
4.1	Serverová aplikace	59
4.1.1	Testování programátorem	59
4.1.2	Integrační testování	59
4.1.3	Zátěžové testování	59
4.1.4	Akceptační testování	59
4.2	Mobilní aplikace	59
4.2.1	Testování programátorem	59
4.2.2	Testování funkcionalit	59
4.2.3	Integrační testování	60
4.2.4	Testování systémové integrace	60
4.2.5	Zátěžové testování	60
4.2.6	Akceptační testování	60
4.2.7	Testování použitelnosti	60
4.2.8	Komparativní testování	60
	Závěr	61
	Literatura	63
A	Seznam použitých zkratk	65
B	Instalační a uživatelská příručka	67
C	Obsah přiloženého CD	69

Seznam obrázků

2.1	Diagram nasazení serverové aplikace	36
2.2	Třídní diagram serverové aplikace	37
2.3	Diagram nasazení mobilní aplikace	50
2.4	Třídní diagram mobilní aplikace	51

Seznam tabulek

Úvod

Když jsem se v květnu roku 2011 začal zabývat diplomovou prací, neměl jsem v tom, co budu dělat, vůbec jasno. Jedním z mála tehdejších cílů bylo vytvoření aplikace, ze které bude mít co největší počet lidí užitek. V rámci bakalářské práce jsem se zabýval doménou navigace studentů po Fakultě elektrotechnické ČVUT [8], proto jsem, mezi jinými tématy, zvažoval i pokračování v obdobné doméně na Fakultě informačních technologií – jedná se o oblast, ve které lze studentům přinést mnoho užitečného.

Nejprve jsem zkontaktoval pana docenta Vitvara, kterého téma oslovilo a začali jsme společně tvořit zadání diplomové práce. Dohodli jsme se na struktuře práce, použitých technologiích a některých dalších detailech. Mezi požadavky se objevila tvorba mobilní aplikace pro přístup k informacím, těmi se pan docent nezabývá, proto mi doporučil další pokračování práce pod panem inženýrem Havrylukem, jehož jsou primární doménou. Této nabídky jsem využil.

Shodná doména bakalářské a diplomové práce se potvrdila být prospěšnou – ačkoliv jsem z bakalářské práce v diplomové mimo získaného přehledu o doméně nakonec nic jiného nevyužil, právě ten se ukázal být jako velmi přínosný – od začátku jsem měl přehled o problematických místech a těch, které tehdy nebylo možné realizovat, a mohl se na ně zaměřit a vyřešit tentokrát podstatně lépe. Bakalářská práce se zabývala pouze navigací do určitého bodu, diplomová jde podstatně více do hloubky i do šířky a přináší celého průvodce postaveného na solidních základech.



Popis problému, specifikace cíle

Tato kapitola se zabývá problematikou zpracovávanou diplomovou prací, kontextem, do kterého je zasazena, rozvržením práce a končí řešerší již zpracovaných obdobných témat.

1.1 Popis řešeného problému

Ačkoliv máme možnost čerpat z mnoha zdrojů, ne vždy nalezneme to, co hledáme. Důvodů je samozřejmě mnoho: Známe velmi omezenou množinu zdrojů, neumíme se v nich zorientovat, nedokážeme je pro jejich komplexitu zpracovat, nevíme, které zdroje jsou ověřené, aktuální... Nic není samozřejmě ztraceno – může se tu objevit někdo, kdo je zmapuje, zagreguje, přetřídí, ohodnotí a pod jednotným rozhraním poskytne. Přesně tento úkol – umožnit studentům snadný přístup k informacím – je cílem diplomové práce.

Fakulta informačních technologií Českého vysokého učení technického v Praze, stejně tak, jako většina ostatních obdobných institucí, nabízí studentům nepřehledné množství informací. Některé se nacházejí na webových stránkách fakulty (<http://fit.cvut.cz/>), univerzity (<http://www.cvut.cz/>), jiné na serverech edux (<https://edux.fit.cvut.cz/>) a komponenta studia (<https://kos.cvut.cz/>), další pak pod Správou účelových zařízení (<http://agata.suz.cvut.cz/jidelnický/>) nebo třeba na webu Centra informačních a poradenských služeb ČVUT (<http://www.cips.cvut.cz/>).

Zorientovat se ve zdrojích nějakou dobu trvá, v těch nejdůležitějších zpravidla ne markantní, přesto existuje mnoho zdrojů, používaných méně často, na které nemusí potenciální uživatel vůbec narazit – jedním z nich je třeba seznam akcí ČVUT (<http://akce.cvut.cz/>). Velmi by proto pomohl nějaký průvodce, který by dostupné informace seskupil a koherentně poskytl.

1.2 Popis uživatele a jeho potřeb

Práce se zaměřuje především na studenty Fakulty informačních technologií Českého vysokého učení technického v Praze, přesto z ní mohou mít užitek i ostatní zainteresované strany – vyučující, neakademičtí pracovníci, zájemci o studium, návštěvníci. . . Kdokoliv pohybující se po dejvickém kampusu nebo hledající nějakou informaci vztaženou k fakultě.

Cílová skupina uživatelů chce zpravidla všechno a hned, nezabývá se proto rozsáhlými předpisy / návody a k informacím se buď dostane během krátké doby z nějakého vhodného zdroje, nebo vůbec. Vhodným zdrojem může být i to, co by pro většinovou populaci vhodné nebylo – studenti Fakulty informačních technologií mají nadprůměrně rozsáhlé znalosti v oblasti informačních technologií, proto může být vhodným zdrojem i komputerovaná řešení.

K předchozí potřebě se váží i nároky na použitelnost výsledného řešení – ačkoliv jsou studenti informačních technologií na ledač zvyklí a zvládnou tak využívat i nestandardní nástroje, mají na druhou stranu profesionální odpor k vyložení špatným řešením. Na přístupnost není třeba brát u této cílové skupiny zřetel.

Studenti zjišťují informace často až na poslední chvíli – například umístění učebny, ve které mají výuku, zjišťují až několik minut před jejím zahájením. V té době nemají možnost využít plnohodnotný počítač, ale musejí se spolehnout na omezené mobilní zařízení, které zpravidla mívají u sebe.

1.3 Vymezení cílů diplomové práce

Cílem diplomové práce je vytvořit průvodce sloužícího studentům jako jednotné rozhraní pro přístup k nejčastěji používaným / nejdůležitějším informacím a usnadnit jim tak určitou – nestudijskou – oblast studentského života. Práci je vhodné rozdělit do dvou částí – mobilní aplikaci, kterou budou studenti využívat, a serverovou aplikaci, která bude sloužit jako univerzální zdroj dat, i pro mobilní aplikaci.¹

Před započítáním implementace je třeba přibližně určit oblasti a rozsahy informací obsažených v průvodci. Následně je potřeba k požadovaným informacím nalézt vhodné zdroje. Neexistující zdroje je třeba nahradit simulovanými zdroji. Přichází na řadu analýza, návrh a realizace nejprve serverové, poté mobilní aplikace. Testování je vhodné provádět po celou dobu.

Mobilní aplikace by měla být dostupná velkému počtu studentů, měla by tedy být multiplatformní a fungovat (jak plyne z označení) i na mobilních

¹ Pojmy *mobilní* a *serverová aplikace* se vyskytují napříč celou prací, až na výjimky všude reprezentují aplikaci sloužící studentům jako průvodce – *mobilní aplikace* a aplikaci shromažďující data z nestrukturovaných zdrojů a poskytujících je unifikovaným způsobem – *serverová aplikace*. V některých případech pojem *serverová aplikace* značí pro nenalezení lepšího označení pouze jednu z komponent tohoto dříve ustanoveného pojmu, význam je z kontextu zřejmý.

zařízeních. Využití aplikace se předpokládá až v následujících letech, není tedy třeba podchytávat stará a méně schopná zařízení – s jejich podporou se již nepočítá. Specifickým zdrojem poskytovaných informací by měla být navigace.

Serverová aplikace by měla být přístupná pouze administrátorům a uživatelům by sloužilo čistě jen vhodné rozhraní pro získávání požadovaných informací. Informace mají být v systému uloženy dle, na systému nezávislého, vhodného modelu.

Práce nemá za cíl plně obsáhnout všechny zpracovávané domény, tím je kvalitní zpracování základu každé domény, které pak lze dle již hotových částí doplnit do celého, pro praktické nasazení potřebného, rozsahu. Postačuje tedy vytvoření částí odpovídajících reprezentativnímu vzorku.

1.4 Popis struktury DP ve vztahu k vytyčeným cílům

Diplomová práce má být členěna do tří hlavních částí:

- Analýza s návrhem.
- Realizace.
- Testování.

Chronologické uspořádání částí je o trochu komplikovanější – začíná se analýzou, která přechází v návrh, po dokončení se vše realizuje, po celou dobu se průběžně testuje. Po ukončení této iterace se začíná nová, se širším záběrem. Tento postup se opakuje až do vytvoření aplikace.

Ve fázi analýzy a návrhu dochází nejprve k průzkumu zpracovávané domény, vytvoření uživatelských scénářů a vytyčení požadavků na vytvářenou aplikaci, později je navržena struktura aplikace, navržena bezpečnost a jsou voleny použité technologie.

Realizace by měla být pouhým mechanickým převedením navrženého systému do reálného světa, bohužel se v praxi vyskytují neočekávané problémy, které je třeba během realizace vyřešit. Vzhledem k akademickému účelu této práce je vhodné uvést obě řešení – nakonec implementované i to zamýšlené ideální.

Testování je třeba provádět na mnoha úrovních – od kontroly zdrojového kódu, přes integrační testování, testování systémové integrace, zátěžové testování, až po testování použitelnosti.

1.5 Serverová aplikace

Systém, který se skrývá pod pojmem *serverová aplikace* (viz vysvětlení v poznámce 1 na straně 20), je velmi specifický a proto pravděpodobně nemá

odpovídající obdoby. Jeho kruciální část, model reprezentující informace potřebné pro průvodce, ale obdob má velmi mnoho, proto byla rešerše prováděna hlavně v této oblasti.

1.5.1 Rešeršní zpracování existujících ontologií

Doposud bylo vytvořeno velké množství univerzitních ontologií – jedná se totiž o oblíbené téma domácích úkolů předmětů zabývajících se sémantickým webem, z toho ale plyne i druhá častá vlastnost těchto ontologií – nebývají příliš propracované a odzkoušené. Kvalitních ontologií je tedy pouze malý zlomek.

1.5.1.1 Univ-Bench

Univ-Bench (<http://swat.cse.lehigh.edu/projects/lubm/>) je, pravděpodobně, nejznámější ontologií modelující univerzitu – jedná se totiž o ontologii využitou v The Lehigh University Benchmark (LUBM) – nástroji pro srovnávací testování úložišť sémantických dat [5]. Ontologie se snaží být co nejrealističtější, vzhledem ke svému účelu se ale nemůže dále rozvíjet. Autorem je Zhengxiang Pan z Lehigh University. Aktuální verze pochází z roku 2004.

1.5.1.2 Academic Institution Internal Structure Ontology

Academic Institution Internal Structure Ontology (AIISO) (<http://purl.org/vocab/aiiso/schema>) je, pravděpodobně, nejpropracovanější ontologií modelující univerzitu. Ve spojení s Participation (<http://purl.org/vocab/participation/schema>), Friend of a friend (FOAF) (<http://xmlns.com/foaf/0.1/>) a AIISO Roles (<http://purl.org/vocab/aiiso-roles/schema>) popisuje téměř celou doménu univerzity [9]. Ontologie je uvolněna pod licencí Creative Commons (attribute). Autory jsou Rob Styles a Nadeem Shabir z Talis Information Ltd. Aktuální verze pochází z roku 2008.

1.5.1.3 Higher Education Reference Ontology

Higher Education Reference Ontology (HERO) (<http://sourceforge.net/projects/heronto/>) je další ontologií mající za cíl pokrýt celou doménu jakékoliv univerzity. Ontologie je uvolněna pod Adaptive Public License. Autorkou je Leila Zemmouchi-Ghomari. Aktuální betaverze, aktivně vyvíjená, pochází z roku 2012 [13].

1.5.1.4 University Ontology

University Ontology (<http://purl.org/weso/uni>) je univerzitní ontologií vycházející z AIISO, FOAF a An organization ontology (Org). Je uvolněna pod licencí Creative Commons (attribute). Autorem je Jose Emilio Labra Gayo z

výzkumné skupiny WESO na Universidad de Oviedo. Aktuální verze pochází z roku 2011 [4].

1.5.1.5 University Ontology

University Ontology (<http://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html>) náleží do projektu Simple HTML Ontology Extensions (SHOE), který má poněkud širší záběr. Autorem je Jeff Heflin z Lehigh University (SHOE, jako celek, ale patří pod katedru informatiky na University of Maryland). Aktuální verze pochází z roku 2000 a v současné době už není udržovaná [6].

1.6 Mobilní aplikace

Na poli mobilních aplikací sloužících jako průvodce studenta je konkurence podstatně větší, než tomu bylo u serverové. Mnoho jich je dokonce napojených na externí zdroj dat, žádná z objevených ale nepracuje přímo se sémantickým úložištěm.

1.6.1 Rešeršní zpracování existujících mobilních aplikací

Práce zpracovávanou doménou navazuje na bakalářskou práci *Mobilní navigační systém pro FEL ČVUT* [8], ta ale není jediným zdrojem, který byl prozkoumán. Ostatní uvedené aplikace zpravidla směřují jinými směry, proto jim nebude věnováno tolik prostoru.

Jako důsledek relativně nedávného prudkého vzestupu počtu chytrých telefonů se za uplynulé roky objevilo mnoho aplikací zabývajících se podobnými tématy, jako tato práce.² Ne vždy se jedná o studentovy průvodce po univerzitě, problém ale řeší obdobný. Nabídka jejich funkcí bývá vzájemně podobná, proto zde uvedu pouze výběr.

V první části této sekce se budu věnovat pracím obdobným této, později se budou objevovat práce stále více odlišné, přesto ale v některých ohledech inspirativní.

1.6.1.1 Mobilní navigační systém pro FEL ČVUT

Bakalářská práce *Mobilní navigační systém pro FEL ČVUT* [8] je zaměřená především na multiplatformnost a použitelnost. Bylo tedy nutné v některých oblastech (včetně funkcionality) podstoupit určité kompromisy. Aplikace vyhledá místnost na základě zadaného označení a přehledným plánem k ní uživatele navede. Označením může být oficiální název, přezdívka, jméno sídlícího pracovníka, či jiný identifikátor. Dále aplikace vizualizuje body zájmu, jako

²Zpracování těchto aplikací bývá ovšem jiné, jsou zaměřené na užitek a sofistikovanost zpracování zpravidla nehraje roli.

jsou občerstvení, výtahy, toalety. . . Práce poskytuje užitečného pomocníka, po čistě programové stránce ale moc zajímavá není.³

1.6.1.2 Průvodce ČVUT

Nyní již zastaralá mobilní aplikace *Průvodce ČVUT FEL* (<http://lr.czechian.net/j2me/>⁴) vytvořená v J2ME poskytovala vyhledávání místností s následným textovým popisem cesty, případným minimalistickým plánkem a dopravním spojením z frekventovaných lokalit.

1.6.1.3 ČVUT navigátor

Projekt *ČVUT navigátor* (<http://navigator.fit.cvut.cz/>) je tématem několika akademických prací. Vypadá slibně, ale ani v době dokončování této práce není známo více, než cíle vytyčené v době založení projektu.

1.6.1.4 Ostatní průvodci ČVUT

Mezi další, v souvislosti s touto prací méně významné, průvodce patří tištěný *Průvodce prváka po ČVUT* [2] nebo například specializovaný *Průvodce prváka po koleji Podolí* (<http://pruvodce.pod.cvut.cz/>).

1.6.1.5 Vlastní průvodci

Mobilní průvodci univerzitami, tak jako se děje i v jiných oblastech mobilních aplikací, se v poslední době začínají velmi rozmáhat. Velké množství univerzit nabízí svým studentům na vlastní půdě vytvořené aplikace, ať již studenty, nebo výpočetními centry. Tyto aplikace bývají šité na míru potřebám univerzit, na druhou stranu ale nemusí být tolik odladěné, jako to bývá u větších projektů. Nabídka funkcionalit sahá od minimalistických až po velmi rozsáhlé. S podporovanými platformami je to podobné – od jedné až po univerzální aplikace.

1.6.1.6 campusM

Univerzity jako Imperial College London, London School of Economics, University College London, Trinity College Dublin a pár desítek dalších využívají komerční aplikace z dílen společnosti *campusM* (<http://www.campusm.com/>).

³Cílem této, diplomové, práce je naopak vytvoření robustní ontologie a samotná mobilní aplikace slouží pouze jako ukázka implementace za využití moderních mobilních technologií, takže ke kompromisům dojde v rozdílných oblastech. Návaznost na bakalářskou práci se bude projevovat převážně v poučení se z předchozích chyb ve shodné zpracovávané doméně.

⁴Aplikace *Průvodce ČVUT FEL* byla, pravděpodobně kvůli neaktualizovanému obsahu a celkovému nedokončení, stažena a nyní již není veřejně dostupná. Nachází se na přiloženém Compact Disc (CD) v adresáři *data/Průvodce ČVUT FEL*.

Řešení je nabízeno pro různé mobilní platformy a nabízí následující funkcionality:

- Vyhledání cesty.
- Lokalizace přátel.
- Novinky a události.
- Upozornění studentů.
- Integrace se systémem knihovny.
- Integrace s univerzitním adresářem.
- Přístup k rozvrhům.
- Informace o předmětech.

1.6.1.7 guidebook

guidebook (<http://guidebook.com/>) je typickým představitelem aplikace pro podporu určité události v širším slova smyslu – může sem spadat i průvodce školou. Ve webovém rozhraní pořadatel událost vytvoří a upraví podle specifických potřeb, na základě toho jsou vytvořeny aplikace pro různé mobilní platformy a ty pak jsou nabízeny návštěvníkům dané události. *guidebook* nabízí obdobné funkcionality výše uvedeným a je kompletně zdarma pro omezený počet uživatelů. Aplikace je pro diplomovou práci inspirativní záběrem a profesionalitou zpracování.

1.6.1.8 Ostatní, neuniverzitní, průvodci

V této široce zaměřené podpodsekci zmíním hlavně průvodce z jednotlivých akcí – ti totiž představují nejmohutnější podmnožinu. V poslední době se stává, hlavně na technicky zaměřených odborných konferencích,⁵ zvykem poskytnout účastníkům mobilní aplikaci disponující informacemi o konferenci – navozuje to profesionální dojem, přináší více možností (ankety, upomínky...) a jsou tu mimo jiné i taková pozitiva, jako možnost aktualizace v případě nenadálé události. Další mobilní průvodci se vyskytují například na hudebních festivalech nebo po městech.

⁵Na těchto konferencích má většina účastníků u sebe zařízení schopné s touto aplikací pracovat.

1.6.1.8.1 Příklady funkcionalit: Žádná z nalezených aplikací nenabízí všechny níže uvedené funkcionality, poměrně velká část se k tomu ale blíží.

- Registrace účastníků.
- Zobrazování harmonogramu.
- Sestavení vlastního programu s upomínkami.
- Zobrazování plánů.
- Zobrazování informací o přednášejících, vystavovatelích, účinkujících. . .
- Zobrazování doplňujících materiálů (slidy a podrobnosti k přednáškám, texty písní u koncertů. . .).
- Interakce se sociálními sítěmi.
- Získávání zpětné odezvy (otázky přednášejícím, ankety. . .).
- Vytvoření prostoru pro reklamu sponzorů.

1.6.1.8.2 Příklady aplikací: Po úvaze jsem se rozhodl odkázat pouze na několik reprezentantů – aplikací je velmi mnoho a vzhledem k podobnosti by nemělo přínos uvádět všechny. Jako ukázkou předkládám jednu svobodnou a několik komerčních aplikací:

iosched (<http://code.google.com/p/iosched/>) je svobodnou aplikací vyvíjenou pro konferenci Google I/O. Nabízí všechny výše uvedené funkcionality vyjma: registrace, sestavování vlastního programu a prostoru pro reklamu, což má svá opodstatnění. Aplikace je napsána v Javě a je určena pro běh v mobilních telefonech se systémem Android. *iosched* je pro diplomovou práci inspirativní záběrem zpracování a možností poučení se z otevřeného zdrojového kódu.

Google Maps Floor Plans (<http://maps.google.com/help/maps/floorplans/>) je dalším, v tomto případě již nesvobodným, nástrojem společnosti Google poskytujícím veřejnosti možnost tvorby plánů vnitřních prostor budov. Tyto plány budou v budoucnu – *Google Maps Floor Plans* je stále ve vývoji – zaneseny do mapových podkladů *Google Maps for Android*. Aplikace je tedy pro diplomovou práci inspirativní hlavně v oblasti navigace uvnitř budov.

Analýza a návrh implementace

Analýza s návrhem probíhaly nezávisle na dvou místech – po krátké analýze celku a určení vzájemných komunikačních rozhraní byly provedeny analýza s návrhem serverové aplikace a až po jejím dokončení začaly analýza s návrhem mobilní aplikace. V průběhu času jsem se k analýze a návrhu ještě několikrát vrátil a vylepšil / rozšířil to, co se ne vždy na poprvé povedlo.

Analýza se skládala z identifikace potřeb uživatelů, vytvoření případů užití, ze kterých byly odvozeny funkční požadavky, ke kterým byly doplněny nefunkční požadavky. Poté byl vytvořen návrh řešení, zapracován návrh bezpečnosti a celý koncept byl podložen návrhem nasazení. Nakonec byly voleny technologie.

2.1 Serverová aplikace

Začal jsem pracovat nejprve na serverové aplikaci – jednak na ní měla být ta mobilní závislá (sama na žádné jiné části práce závislá není) a také má užší vazbu ke zpracovávané doméně.

2.1.1 Potřeby uživatelů

Tato část se věnuje potřebám uživatelů serverové aplikace, zejména uživatelů rozhraní, přes které poskytuje informace. Potřeb je mnoho – proto zde uvedu pouze reprezentativní vzorek.

2.1.1.1 Potřeba zjistit informaci

Potřeba zjistit informaci je základní potřebou, kvůli které se mají uživatelé obracet na systém. Tato potřeba může vzniknout z velkého množství příčin – od hledání informací spjatých s průchodem studiem, až například po zjišťování jídelního lístku v menze.

2.1.1.2 Potřeba ověřit informaci

Často je cílová skupina uživatelů v situaci, kdy má požadovanou informaci, není si jí ale jistá a potřebuje ji ověřit / vyvrátit. Bez jistoty pravdivosti informace je její hodnota podstatně nižší.

2.1.1.3 Potřeba ověřených informací

Velmi podstatnou potřebou je potřeba ověřených informací. Ačkoliv je název velmi podobný té předchozí, jedná se o dvě diametrálně odlišné potřeby, předchozí se bez naplnění této neobejde. Jakmile jsou informace poskytovány, je nutné, aby byly ověřené – pravdivé.

2.1.1.4 Potřeba aktuálních informací

Potřeba aktuálních informací opět souvisí s již zmíněnými potřebami – sebepravdivější jídelní lístek z předchozího dne mi při výběru dnešního oběda bohužel moc nepomůže.

2.1.1.5 Potřeba kompletních informací

Někdy nestačí jen pravdivé a aktuální informace, naopak vědomí záruky pravdivosti a aktuálnosti může i uškodit – pokud totiž informace nejsou kompletní, může dojít k jejich misinterpretaci a pod vlivem záruk nedojde ke kritickému posouzení. Proto je zde i potřeba kompletních informací.

2.1.1.6 Potřeba nalézt, co nehledám

Na první pohled tato potřeba nedává smysl, v praxi se ale často vyskytují informace, které bychom měli znát a o jejich existenci nemáme ani ponětí. Je proto důležité ve vhodné chvíli sdělit i nepožadované informace, které jsou s velkou pravděpodobností pro adresáta důležité.

2.1.2 Případy užití

Případy užití (*use cases*) jsou takové sekvence kroků, které reprezentují danou činnost prováděnou aktérem na systému a vedou od jejího zahájení až po naplnění.

V serverové aplikaci byli identifikováni celkem čtyři hlavní aktéři:

- Uživatel – student / softwarová komponenta (*mobilní aplikace*).
- Správce – administrativní pracovník fakulty.
- Plánovač – softwarová komponenta.
- Zdroj – server poskytující informace.

Roli správce by šlo dále dělit na specifitější role, vzhledem k velmi nízkému rozsahu činností každé z nich (zpravidla jedna činnost), je vše zahrnuto pod tímto souhrnným názvem. Pod plánovačem je zase zahrnut aktér čas.

Uživatel se pouze dotazuje nad systémem a zjišťuje informace. Správce manuálně upravuje informace v systému, může vynutit aktualizaci a jinak spravuje systém. Plánovač automaticky obsluhuje aktualizace obsažených informací. Zdroj na vyžádání poskytne dané informace.

2.1.2.1 UC-S-1: Vyžádání informací

Aktéři:

- Uživatel.

Systém:

- Serverová aplikace.

Vstupní podmínky:

- Uživatel zná jazyk dotazovacího rozhraní systému.
- Uživatel má přístup k dotazovacímu rozhraní systému.

Hlavní scénář:

- Uživatel vytvoří dotaz.
- Uživatel odešle dotaz na rozhraní systému.
- Systém přijme dotaz.
- Systém zpracuje dotaz – nalezne požadované informace.
- Systém odešle odpověď.
- Uživatel přijme odpověď.
- Uživatel zpracuje odpověď.

Alternativní scénáře:

- V případě neplatného dotazu je navraceno chybové hlášení.
- V případě chyby serveru je navraceno chybové hlášení.

Poznámky:

- V případě nenalezení informace je navracena prázdná množina.
- Pokud je uživatelem softwarová komponenta, student ji obsluhující nemusí znát jazyk dotazovacího rozhraní – může s komponentou komunikovat jiným způsobem a ta si požadavky do požadovaného jazyka převádět.

2.1.2.2 UC-S-2: Aktualizace informací

Aktéři:

- Plánovač.
- Zdroj.

Systém:

- Serverová aplikace.

Vstupní podmínky:

- Plánovač je řádně nastaven a spuštěn.
- Nastal čas aktualizace určitých informací.

Hlavní scénář:

- Plánovač zadá systému požadavek na aktualizaci určitých informací.
- Systém přijme požadavek.
- Systém dá zdroji požadavek na informace.
- Zdroj zpracuje požadavek a informace vrátí.
- Systém přijme informace.
- Systém zpracuje informace.
- Systém odešle plánovači potvrzení.

Alternativní scénáře:

- V případě chyby zdroje je navraceno chybové hlášení.
- V případě chyby zpracování informací je navraceno chybové hlášení.

Poznámky:

- Případné chyby jsou zaznamenány do logu.
- Aktualizaci může obdobným způsobem vynutit i správce.
- Zdrojů určitých informací může být i více a vzájemně se mohou doplňovat / ověřovat.

2.1.2.3 UC-S-3: Správa informací

Aktéři:

- Správce.

Systém:

- Serverová aplikace.

Vstupní podmínky:

- Správce má přístup do systému.
- Správce zná jazyk rozhraní správy systému.

Hlavní scénář:

- Správce vytvoří dotaz.
- Správce se připojí do systému.
- Správce odešle dotaz.
- Systém přijme dotaz.
- Systém zpracuje dotaz.
- Systém odešle odpověď.
- Správce přijme odpověď.
- Správce se odpojí ze systému.
- Správce zpracuje odpověď.

Poznámky:

- Případné chyby jsou zaznamenány do logu.

2.1.3 Funkční požadavky

Funkčními požadavky (*functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na jednotlivé funkcionality systému. Ne všechny požadavky budou nakonec implementovány, pouze jejich reprezentativní vzorek, dle kterého bude možné systém o ty ostatní rozšířit. Pro serverovou aplikaci mezi ně patří především:

1. Aplikace bude čerpat informace o názvech budov z předstíraného adaptéru.⁶
2. Aplikace bude čerpat informace o umístění budov z předstíraného adaptéru.⁷
3. Aplikace bude čerpat informace o názvech místností z předstíraného adaptéru.⁸
4. Aplikace bude čerpat informace o umístění místností z předstíraného adaptéru.⁹
5. Aplikace bude čerpat seznam vyučujících z KOSapi (<http://kosapi.fit.cvut.cz/>).¹⁰
6. Aplikace bude čerpat seznam studentů z KOSapi (<http://kosapi.fit.cvut.cz/>).¹¹
7. Aplikace bude čerpat seznam neakademických pracovníků z předstíraného adaptéru.
8. Aplikace bude čerpat informace o rozvrzích místností z KOSapi.¹²
9. Aplikace bude čerpat informace o rozvrzích vyučujících z KOSapi (<http://kosapi.fit.cvut.cz/>).¹³

⁶Vynecháno. Implementováno na straně *mobilní aplikace*.

⁷Vynecháno. Implementováno na straně *mobilní aplikace*.

⁸Vynecháno. Implementováno na straně *mobilní aplikace*.

⁹Vynecháno. Implementováno na straně *mobilní aplikace*.

¹⁰Nahrazen zdroj za předstíraný adaptér – žádný seznam vyučujících obsahující identifikátory (uživatelská jména) nebyl nalezen.

¹¹Nahrazen zdroj za předstíraný adaptér – z důvodů konzistence se zdrojem vyučujících.

¹²Vynecháno – projektu KOSapi nebyl dosud umožněn přístup do komponenty studia. Implementováno na straně *mobilní aplikace*.

¹³Vynecháno – projektu KOSapi nebyl dosud umožněn přístup do komponenty studia. Implementováno na straně *mobilní aplikace*.

10. Aplikace bude čerpat informace o kancelářích vyučujících z Usermap ČVUT (<http://usermap.cvut.cz/>).
11. Aplikace bude čerpat informace o kontaktech vyučujících z Usermap ČVUT (<http://usermap.cvut.cz/>).
12. Aplikace bude čerpat informace o rozvrzích studentů z KOSapi.¹⁴
13. Aplikace bude čerpat informace o kontaktech studentů z Usermap ČVUT (<http://usermap.cvut.cz/>).
14. Aplikace bude čerpat informace o kancelářích neakademických pracovníků z Usermap ČVUT (<http://usermap.cvut.cz/>).
15. Aplikace bude čerpat informace o kontaktech neakademických pracovníků z Usermap ČVUT (<http://usermap.cvut.cz/>).
16. Aplikace bude čerpat informace o akcích ČVUT z Kalendáře akcí ČVUT (<http://akce.cvut.cz/>).
17. Aplikace bude čerpat informace o harmonogramu akademického roku z předstíraného adaptéru (na základě <http://fit.cvut.cz/>).¹⁵
18. Aplikace bude čerpat informace o datech MAR budovy T9.¹⁶
19. Aplikace bude čerpat informace o datech MAR kolejí Orlík.¹⁷
20. Aplikace bude čerpat informace o datech MAR Masarykovy koleje.¹⁸
21. Aplikace bude čerpat informace o jídelničkách menz z Jídelníčků menz (<http://agata.suz.cvut.cz/jidelnicky/>).
22. Aplikace bude čerpat informace o otvírací době menz z Jídelníčků menz (<http://agata.suz.cvut.cz/jidelnicky/>).¹⁹
23. Aplikace bude čerpat informace o otvírací době NTK z Národní technické knihovny (<http://www.techlib.cz/>).²⁰
24. Aplikace bude čerpat informace o otvírací době Vydavatelství průkazů (<http://ke.customer.decent.cz/a021/mon/wc-mon.php?co=2>).²¹

¹⁴Nahrazen zdroj za libovolný zdroj ve formátu iCalendar – projektu KOSapi nebyl dosud umožněn přístup do komponenty studia.

¹⁵Vynecháno. Implementováno na straně *mobilní aplikace*.

¹⁶Jednání o udělení přístupu nedopadla – vyskytly se blíže nespecifikované technické komplikace na straně technickoprovozních služeb Fakulty architektury.

¹⁷Vynecháno. Implementováno na straně *mobilní aplikace*.

¹⁸Vynecháno – data poskytována pouze pod licencí, kterou nebylo možné naplnit.

¹⁹Vynecháno. Implementováno na straně *mobilní aplikace*.

²⁰Vynecháno. Implementováno na straně *mobilní aplikace*.

²¹Vynecháno. Implementováno na straně *mobilní aplikace*.

25. Aplikace bude čerpat informace o počtu lidí ve frontě ve Vydavatelství průkazů (<http://ke.customer.decent.cz/a021/mon/>).²²
26. Vytvořte / nalezněte / sestavte z dílčích částí ontologii reprezentující výše uvedené informace.
27. Informace získané z jednotlivých zdrojů převádějte dle výše požadované ontologie pro další zpracování do RDF.
28. Aplikace umožní automatické aktualizace informací.
29. Aplikace umožní manuální aktualizace informací.
30. Aplikace umožní manuální úpravy informací.
31. Aplikace bude logovat chyby.
32. Aplikace bude ukládat sémantické informace do databáze.
33. Aplikace umožní vykonávat dotazy nad uloženými daty.

2.1.4 Nefunkční požadavky

Nefunkčními požadavky (*non-functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na dílo jako celek, nikoliv na jednotlivé funkcionality. Pro serverovou aplikaci mezi ně patří především:

1. Aplikaci naprogramujte v jazyce *JavaScript*.
2. Pro běh serveru využijte *Node.js*.
3. Pro dotazování využijte dotazovací jazyk *SPARQL*.
4. Sémantická data ukládejte do takového úložiště, aby nad ním bylo možné vykonávat dotazy přímo (případně přes nezávislého prostředníka) a nebylo nutné přistupovat přes vlastní aplikaci.
5. Práce nemusí nutně obsáhnout celou zpracovávanou doménu, je ale žádané, aby kvalitně zpracovala hlavní scénáře využití a nebránila dalšímu rozšiřování.
6. Zveřejněte aplikaci a její data pod svobodnými licencemi z rodin *GNU*, *Creative Commons* a kompatibilními, aby byl umožněn další rozvoj nezávislý na původním autorovi.

²²Vynecháno.

2.1.5 Návrh ontologie

Tvorbě ontologie bylo věnováno velmi dlouhé období – neměl jsem s tímto oborem předchozí zkušenosti a vytvoření kvalitní ontologie považuji jako nutný předpoklad pro následné pokračování – na špatných základech nelze vybudovat kvalitní systém.

Zpočátku jsem vytvářel ontologii od základů vlastní, s přibývajícimi zkušenostmi jsem si ale uvědomil, že to nebyl dobrý nápad – ve světě sémantického webu je jednou z nejdůležitějších priorit vzájemná interoperabilita – s každou novou ontologií je tak vzájemné mapování informací čím dál tím těžší. V té době jsem veškerou svou dosavadní práci zahodil a začal znovu.

Nejprve jsem se vrátil k výsledkům rešeršního zpracování ontologií a vybral z nich tu, která se mi zdála po v mezičase nabitých zkušenostech nejlepší – AIISO. Tato ontologie je již předurčená k využití s ontologiemi Participation, FOAF a AIISO Roles. Ontologie jsou to již zaběhlé, takže problém s interoperabilitou pozbyl platnosti.

Dalším bodem zvratu bylo zjištění, jak složité může být nalezení styčných bodů mezi obecnou ontologií a konkrétní situací. Bylo třeba v některých případech nalézt další vhodné ontologie reprezentující doposud nepokryté oblasti, případně již použité ontologie rozšířit (zpravidla je použita licence Creative Commons, takže je rozšiřování možné). Příkladem může být reprezentace akcí ČVUT – žádná z použitých ontologií nemodeluje události. Abych se dodržel kurzu ve směru interoperability, bylo třeba nalézt vhodnou ontologii. Nabízelo se jich několik, nakonec byla použita Linking Open Descriptions of Events (LODE), právě kvůli svému zastřešení obdobných ontologií.

Následuje seznam zvolených ontologií, pro specifické případy se ale počítá s jeho dalším rozšiřováním:

- AIISO (<http://purl.org/vocab/aiiso/schema>).
- Participation (<http://purl.org/vocab/participation/schema>).
- FOAF (<http://xmlns.com/foaf/0.1/>).
- AIISO Roles (<http://purl.org/vocab/aiiso-roles/schema>).
- LODE (<http://linkedevents.org/ontology/>).
- OWL-Time (<http://www.w3.org/TR/owl-time/>).

2.1.6 Návrh architektury

Nejprve byl vytvořen návrh architektury (*architecture design*) aplikace, který byl později rozpracován do návrhu nasazení *deployment design*.

Identifikovány byly následující základní struktury systému:

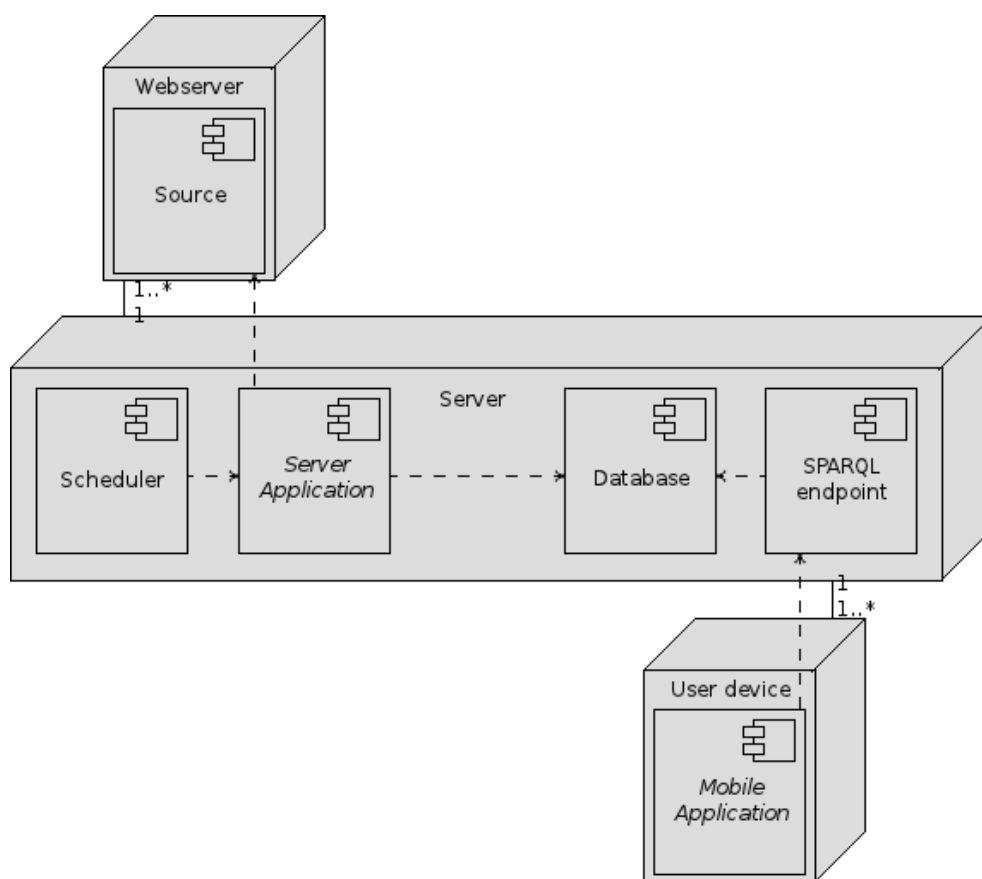
- *Serverová aplikace*.

2. ANALÝZA A NÁVRH IMPLEMENTACE

- Plánovač.
- Databáze.
- SPARQL endpoint.

Pod pojmem *serverová aplikace* se skrývá aplikace, která získává data přímo ze zdrojů, z nich vybírá informace, ty strukturuje a ukládá do databáze. Plánovač je reprezentován aplikací, která ve vhodné časy vyžádá na serveru aktualizaci informací. SPARQL endpoint je aplikace, která na požadavek v dotazovacím jazyku vrátí odpovídající data z databáze. Databáze je úložištěm informací.

Nasazení bylo nakonec navrženo, jak je vidět na obrázku 2.1, umístěním všech prvků serverové aplikace na jediný server, který získává informace z prostředí Internetu a zároveň poskytuje informace uživatelům.



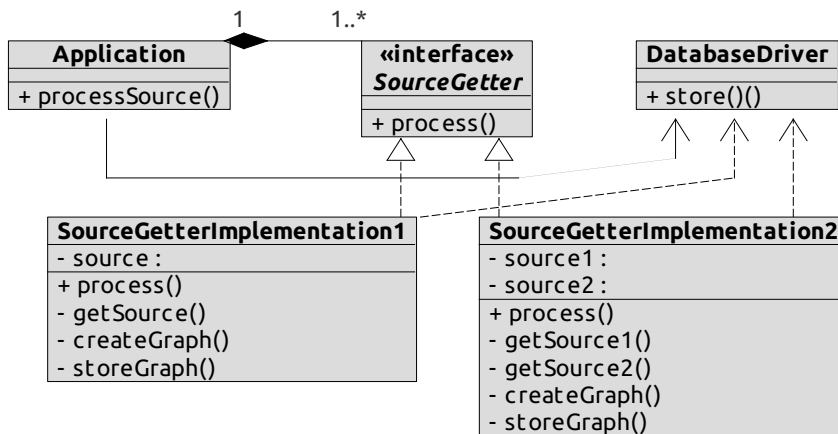
Obrázek 2.1: Diagram nasazení serverové aplikace

2.1.7 Návrh řešení

Návrh (*design*) byl prováděn po rozvržení architektury a za přispění volby použitých technologií.

Pro tři ze čtyř dříve identifikovaných struktur systému jsem našel vhodné hotové řešení (viz část 2.1.9 na straně 39) – plánovač je tvořen cronem, jako databáze je použita MongoDB a SPARQL endpoint poskytuje rdfstore. Zbývá tedy doimplementovat pouze *serverovou aplikaci*.

Na obrázku 2.2 je znázorněn třídní diagram (*class diagram*) reprezentující strukturu serverové aplikace. Metoda **process()** má za úkol stažení dat z příslušných zdrojů, jejich zpracování, převedení do sémantických grafů a uložení do databáze. V návrhu byl použit návrhový vzor strategie (*strategy*).



Obrázek 2.2: Třídní diagram serverové aplikace

2.1.8 Návrh bezpečnosti

Po zběžném návrhu aplikace byly modelovány hrozby. Nejprve jsem aplikaci dekomponoval na důvěryhodné a nedůvěryhodné komponenty systému a určil mezi nimi hranice důvěry. Poté jsem identifikoval hrozby. Nakonec byly identifikovány zranitelnosti. Vymodelované hrozby byly následně snižovány.

2.1.8.1 Dekompozice

Byly identifikovány následující komponenty (obdobné s těmi z návrhu architektury):

- *Serverová aplikace.*
- Plánovač.
- Databáze.

- SPARQL endpoint.
- Uživatel.
- Zdroj informací.

Plně důvěřovat můžu pouze serverové aplikaci a plánovači – mají to být vlastní jednoduché aplikace. Dále lze do určité míry důvěřovat databázi a SPARQL endpointu – aplikace budou provozovány pod vlastní správou. Zdroje informací a uživatelské aplikace jsou naopak nedůvěryhodné – jsou plně v režii třetích stran. Hranice důvěry jsou nicméně mezi všemi komponentami.

2.1.8.2 Identifikace hrozeb

U komponenty *serverová aplikace* byly identifikovány následující hrozby:

- Podvržení nepravdivých dat.
- Přetížení komponenty velkým počtem požadavků.

U komponenty *databáze* byly identifikovány následující hrozby:

- Přetížení komponenty velkým počtem požadavků.
- Podvržení nepravdivých dat.

U komponenty *SPARQL endpoint* byly identifikovány následující hrozby:

- Přetížení komponenty velkým počtem požadavků.
- Podvržení identity uživatele.
- Únik informací.

U komponenty *plánovač* nebyly identifikovány, vzhledem k jednoúčelovému zaměření, žádné hrozby.

2.1.8.3 Identifikace zranitelností

Všechny výše uvedené hrozby mohou vyústit ve zranitelnosti, nejvyšší prioritu mají ty nejpravděpodobnější – na hranicích důvěry směrem k uživatelům a zdrojům dat. S hrozbou můžeme naložit následovně:

- Ponechat v programu.
- Odstranit problém.
- Odstranit hrozbu.

Preferovaná je zpravidla poslední možnost.

2.1.8.4 Důsledky

- Je třeba důsledně ošetřovat vstupy (aby nedošlo k uložení škodlivých dat do databáze, aby na SPARQL endpointu nebyly prováděny jiné operace, než dotazovací...).
- Je třeba vše ve výchozím nastavení vše zakázat a k potřebným úkonům udělit výjimky (pouze dotazy nad SPARQL endpointem, přístup jen k některým zdrojům ze serverové aplikace...).
- Je třeba zamezit odepření služeb (přetížením SPARQL endpointu, zahlcením databáze...).

2.1.9 Použité technologie

Mnoho použitých technologií bylo dáno zadáním práce, byl mi tak do značné míry usnadněn jejich výběr. Na druhou stranu nejsou některé z technologií příliš ověřené, zdokumentované a neexistují k nim příklady, takže by často bývala práce s jinými snazší.

Celá serverová aplikace je napsána v JavaScriptu, zprovozněna na Node.js, datovým úložištěm se stalo MongoDB.

2.1.9.1 JavaScript

Využití JavaScriptu bylo, pro svoje pravděpodobné vzrůstající nasazení v budoucnosti a své specifické vlastnosti, dáno jako nefunkční požadavek.

2.1.9.2 Node.js

Node.js (<http://nodejs.org/>) je platforma postavená nad běhovým prostředím JavaScriptu z webového prohlížeče Google Chrome. Slouží ke tvorbě rychlých, škálovatelných síťových aplikací. Node.js využívá událostmi řízený neblokující I/O model, který ho činí nenáročným a efektivním, vhodným pro datově náročné *real-time* aplikace běžící v distribuovaném prostředí [12].

Použití Node.js bylo vynuceno nefunkčním požadavkem, jedná se ale o velmi cennou zkušenost – je novou technologií a ačkoliv je jeho dokumentace na slušné úrovni, není zatím vytvořena dostatečně velká báze příkladů a nejlepších praktik.²³

2.1.9.3 MongoDB

MongoDB (<http://www.mongodb.org/>) je škálovatelnou, vysoce výkonnou, otevřenou NoSQL databází napsanou v C++ [11].

²³Takže jsem se k dobrým praktikám musel postupně propracovat a při té příležitosti se toho hodně naučil. Navíc, zdá se, se budou nabitě vědomosti v budoucnu hodit.

Databázi jsem volil z dalších sémantických / NoSQL databází (zaměření dáno požadavkem na použití nových inovativních technologií), zvažovány byly mimo jiné i CouchDB (<http://couchdb.apache.org/>), Neo4j (<http://neo4j.org/>) a Openlink Virtuoso (<http://virtuoso.openlinksw.com/>).

Poslední jmenovaná databáze byla na základě předchozích zkušeností vybrána nejdříve, později se ale ukázalo, že jsou její správa i obsluha pro konkrétní použití těžkopádné, takže jsem nakonec přešel k podstatně vhodnější databázi, MongoDB, kterou lze navíc snadno využít v kombinaci s JavaScriptovým RDF úložištěm rdfstore (viz část 2.1.10.6).

2.1.9.4 cron

Cron je démonem vykonávajícím naplánované úlohy. Název neoznačuje konkrétní implementaci, ale používá se pro označení programu splňujícího určité specifikace (jak je u Unixových nástrojů zvykem). Byla vybrána implementace, jejímž autorem je Paul Vixie – jedná se o verzi využitou například v GNU/Linuxu a BSD (odkud pochází).

Cron byl vybrán pro svou vhodnost pro daný účel, plánovat, a pro svoji prověřenost.

Celé prostředí serverové aplikace je umístěno v serverové edici GNU/Linuxové distribuce Ubuntu (<http://www.ubuntu.com/>) a ta je pro snadnou přenositelnost nainstalována do kontejneru virtuálního stroje virtualizačního řešení VirtualBox (<http://www.virtualbox.org/>). Technologie byly vybrány na základě výborných předchozích zkušeností.

2.1.10 Použité knihovny

Mimo standardního JavaScriptu lze v Node.js využívat i specifická rozšíření a množství modulů třetích stran. Z těch jsem po zvážení z různých příčin využil:

2.1.10.1 node-date-utils

Node-date-utils (<https://github.com/JerrySievert/node-date-utils>) je mikro frameworkem doplňujícím funkcionalitu JavaScriptového objektu Date. Nástroj byl vybrán z potřeby snadno formátovat čas.

Zvažováno bylo i použití obdobného nástroje node-dateformat (<https://github.com/felixge/node-dateformat>), ten je ale svou nabídkou funkcionalit pro projekt méně vhodný. Oba moduly jsou stále ve vývoji.

2.1.10.2 express

Express (<http://expressjs.com/>) je renomovaným frameworkem pro tvorbu webových aplikací. Je založený na middleware frameworku connect (<http://expressjs.com/>).

`//www.senchalabs.org/connect/`).

Express byl vybrán jako ověřené řešení pro vytvoření struktury webové aplikace – nejenže v sobě zahrnuje mnoho dobrých praktik, ke kterým bych jen stěží došel vlastním zkoumáním, ale přináší zároveň i markantní zjednodušení struktury kódu aplikace.

2.1.10.3 node-icalendar

Node-icalendar (<https://github.com/tritech/node-icalendar>) je knihovnou pro práci s formátem iCalendar. Je stále ve vývoji, nicméně množina poskytovaných funkcionalit je pro tuto práci dostatečná.

Zvažovány byly i další knihovny, například node-ical (<https://github.com/peterbraden/node-ical>), ty ale neimplementují dostatek funkcionality potřebné pro tvorbu aplikace.

2.1.10.4 node-iconv

Node-iconv (<https://github.com/bnoordhuis/node-iconv>) je knihovnou pro obsluhu systémového příkazu iconv sloužícího k převádění kódování. Knihovna byla vybrána pro bezpečnou implementaci převodu kódování.

2.1.10.5 node-jquery

Node-jquery (<https://github.com/coolaj86/node-jquery>) je *wrapperem* umožňujícím využít renomovanou JavaScriptovou knihovnu jQuery (<http://jquery.com/>) v prostředí Node.js.

jQuery bylo vybráno pro usnadnění zpracování externích zdrojů dat – nabízí prověřené způsoby selekce, traverzování a manipulace s DOM.

Node-jquery interně využívá modul jsdom (<https://github.com/tmpvar/jsdom>), který mu poskytuje JavaScriptovou implementaci DOM.

2.1.10.6 rdfstore

Rdfstore (<https://github.com/antoniogarrote/rdfstore-js>) je JavaScriptovou implementací RDF grafového úložiště s podporou pro SPARQL. Data mohou být (a v implementaci DP jsou) persistentně ukládána prostřednictvím Node.js modulu-ovladače node-mongodb (<https://github.com/christkv/node-mongodb-native>) do NoSQL databáze MongoDB (<http://www.mongodb.org/>). Modul je stále ve vývoji.

Právě pro implementaci manipulace s RDF, grafové úložiště a podporu pro SPARQL byl rdfstore vybrán jako základ pro SPARQL endpoint a důležitá součást *serverové aplikace*.

2.1.11 Použité nástroje

Pro tvorbu serverové aplikace byly vybrány následující nástroje:

2.1.11.1 Kate

Pro implementaci serverové aplikace, stejně jako pro mnoho dalších činností, byl vybrán textový editor Kate (<http://kate-editor.org/>) z prostředí GNU/Linuxového desktopového prostředí KDE. Kate nabízí zvýrazňování syntaxe všech v diplomové práci použitých formátů, poskytuje velmi užitečnou práci s regulárními výrazy, umožňuje pracovat s mnoha soubory najednou a, co je asi nejpodstatnější, plně se integruje do (mnou používaného) KDE, takže ním například lze prostřednictvím KDE Input/Output (KIO) transparentně pracovat se vzdálenými soubory. Kate neumožňuje automatické doplňování kódu, nabízí ale již použitá slova kdekoliv v dokumentu, což je vlastnost v mnoha ohledech srovnatelná. Kate byl zvolen i na základě velice kladných zkušeností pro vytvoření zdrojového kódu a konfiguračních souborů.

2.1.11.2 Kompare

Pro případ potřeby důkladného porovnání souborů byl vybrán Kompare (<http://www.caffeinated.me.uk/kompare/>). Nástroj umožňuje velmi přehledně porovnat mimo jednotlivých souborů i celé jejich adresáře. Kompare neporovnává soubory sám, ale je nadstavbou nad konzolovou aplikací diff či obdobnou. Mimo již popsaných vlastností byl Kompare vybrán i pro svou integraci s KDE.

2.1.11.3 Webové prohlížeče

Pro průzkum zdrojů a jejich struktur byly zvoleny internetové prohlížeče a jejich rozšíření:

- Mozilla Firefox (<http://www.mozilla.org/firefox>) + Firebug (<http://getfirebug.com/>).
- Chromium (<http://chromium.org/>).

2.1.11.4 Protégé

Pro průzkum a práci s ontologiemi byl zvolen editor Protégé (<http://protege.stanford.edu/>). Rozhodnutí bylo učiněno pro nabídku funkcionalit a předchozí zkušenosti.

2.1.11.5 curl

Konzolový nástroj curl z projektu cURL (<http://curl.haxx.se/>) byl zvolen pro snadnou kontrolu cizích zdrojů i vlastní vytvářené aplikace.

2.1.11.6 Git

Pro správu verzí práce byl zvolen nástroj Git (<http://git\discretionary{-}{-}{-}scm.com/>), čímž má být zajištěné pohodlné zálohování, jak ve smyslu možnosti návratu k některé z minulých verzí, tak i ve smyslu vzdáleného duplikování na serveru GitHub (<http://github.com/>). Pozitivním vedlejším efektem tohoto přístupu je možnost sledování vývoje prostřednictvím veřejného repozitáře.

2.2 Mobilní aplikace

Mobilní aplikace má sloužit studentům jako průvodce a měla by spolupracovat se serverovou aplikací. Vzhledem k této závislosti je třeba ji realizovat až po té serverové, ovšem analýza s návrhem by měly probíhat kvůli vzájemné spolupráci v některých fázích souběžně.

2.2.1 Potřeby uživatelů

Následující část se věnuje identifikaci základních potřeb potenciálních uživatelů aplikace, které by jim mohla / měla naplnit – na jejich základě totiž budou hledány uživatelské scénáře a požadavky.

2.2.1.1 Potřeba informací

Pod tuto potřebu shrnuji všechny potřeby nalezené u serverové aplikace (viz část 2.1.1 na straně 27). Jsou to:

- Potřeba zjistit informaci.
- Potřeba ověřit informaci.
- Potřeba ověřených informací.
- Potřeba aktuálních informací.
- Potřeba kompletních informací.
- Potřeba nalézt, co nehledám.

Zejména naplnění poslední potřeby – předložit informaci, o kterou člověk sice nepožádal, ale udělal by to, kdyby ho to bylo napadlo – je od průvodce vítané.

2.2.1.2 Potřeba uživatelské přívětivosti

Vzhledem k předpokládanému využití mobilní aplikace podstatně méně pokročilými uživateli oproti serverové je vhodné, aby tomu byla uzpůsobena. Cílová skupina navíc zpravidla není ochotná číst dlouhé návody a je tak třeba, aby byla aplikace použitelná sama o sobě.

2.2.1.3 Potřeba určit cílovou pozici

Jedna z nejdůležitějších potřeb, které by měl průvodce naplnit, je potřeba určit cílovou pozici. Může to být například:

- Budova, učebna, kancelář, knihovna...
- Menza, hospoda, občerstvení, nápojový automat...
- Pitná voda, toalety, sprchy...
- Výtah, schodiště, nájezd pro kočárek...
- Vchod, únikový východ, hasící přístroj...
- Bankomat, zastávka MHD, parkoviště...
- Informace, vrátnice, šatna, údržba...

Ačkoliv jsou v seznamu uvedené pozice rozděleny do určitých logických celků, je důležité si povšimnout, že existují i jiná jejich, v některých ohledech důležitější, uspořádání:

- Dle jednoznačnosti místa – místnost s jasně daným názvem je oproti libovolnému schodišti jediná.
- Dle jednoznačnosti identifikátoru – stejná místnost může být vyhledána na základě různých kritérií (název, další výuka, popis).
- Dle frekvence hledání – přednáškové místnosti jsou vyhledávány podstatně častěji, než hasící přístroje.
- Dle kritéria priority – vzdálenější menza může mít vyšší prioritu na základě lepší kuchyně, zatímco sebelepší nápojový automat může mít kvůli své větší vzdálenosti prioritu nižší.

Pro úplnost ještě uvádím seznam informací vedoucích k určení cílové pozice:

- Označení místnosti, budovy... (T9:349, TK...).
- Název místnosti, budovy... (Gočár, Národní technická knihovna...).
- Neoficiální název místnosti, budovy... (Bouračka, Modrá menza...).
- Jméno, pokud hledáme kancelář vyučujícího nebo neakademického pracovníka.
- Rozvrh, pokud hledáme místo další výuky nebo zkoušky.
- Zeměpisné souřadnice.

- Poloha všech míst, které přicházejí v úvahu (toalety, bankomaty...) – hledající si vybere.
- Předchozí bod a navíc přibližná aktuální poloha – lze určit nejbližší.

Přehledy rozšiřují poznatky získané při tvorbě bakalářské práce [8].

2.2.1.4 Potřeba určit aktuální pozici

Zvláště, pokud máme mapu nebo jsme v jiné situaci, kdy víme, kam jít, vystává potřeba určit aktuální pozici – bez té totiž nelze určit směr, vzdálenost, a v případě, že hledáme nejbližší místo daného typu, ani cíl cesty.

2.2.1.5 Potřeba nezávislosti

Cílová skupina má díky specifickému životnímu stylu silnou potřebu nezávislosti v mnoha oblastech – pokud jim má být poskytnu aplikace, ani ta je nesmí příliš omezovat – je proto vhodné, aby se jednalo o aplikaci mobilní a aby umožňovala offline provoz, jinak hrozí nemožnost jejího využití v situacích jako je cesta do školy nebo přítomnost na některém z mnoha signálem nepokrytých míst budovy T9.

2.2.2 Případy užití

U případů užití (*use cases*) mobilní aplikace byli identifikováni jediní dva aktéři:

- Uživatel – student.
- Server – SPARQL endpoint.

Dále byly identifikovány tři hlavní případy užití:

- Získání určité (textové) informace.
- Vyhledání (mapové) pozice.
- Vykonání SPARQL dotazu.

2.2.2.1 UC-M-1: Získání jídelníčku

Aktéři:

- Uživatel.
- Server.

Systém:

- Mobilní aplikace.

Vstupní podmínky:

- Uživatel se nachází na patřičném místě systému.
- Systém má připojení k serveru.

Hlavní scénář:

- Uživatel nalezne akci získání jídelníčku.
- Uživatel dá požadavek na získání jídelníčku.
- Systém odešle požadavek na server.
- Server zpracuje požadavek a odešle odpověď.
- Systém přijme odpověď
- Systém zpracuje odpověď.
- Systém vizualizuje jídelníček.

Alternativní scénáře:

- V případě chyby připojení je navraceno chybové hlášení.
- V případě neexistence jídelníčku je navraceno oznámení.

Poznámky:

- Místo získávání jídelníčku může být prováděna libovolná obdobná činnost.
- Pro některé obdobné informace nemusí být vyžadováno připojení k serveru, záleží na době jejich platnosti.

2.2.2.2 UC-M-2: Vyhledání občerstvení

Aktéři:

- Uživatel.

Systém:

- Mobilní aplikace.

Vstupní podmínky:

- Uživatel se nachází na patřičném místě systému.

Hlavní scénář:

- Uživatel zadá do vyhledávacího pole požadavek na občerstvení.
- Uživatel potvrdí vyhledávání.
- Systém vyhledá všechna občerstvení.
- Systém vizualizuje nalezená občerstvení.
- Uživatel zadá požadavek na zobrazení aktuální polohy.
- Systém určí uživatelskou polohu.
- Systém vizualizuje polohu.
- Uživatel si dle poskytnutých informací vybere občerstvení.

Alternativní scénáře:

- V případě neplatného dotazu je navraceno chybové hlášení.
- V případě chyby určování polohy je navraceno chybové hlášení.
- V případě výskytu mimo mapu je navraceno chybové hlášení.

Poznámky:

- Místo vyhledávání občerstvení mohou být vyhledávány i jiné pozice.
- U některých typů dotazů není nutné znát aktuální polohu.
- U některých typů dotazů je navracena jediná pozice.

2.2.2.3 UC-M-3: Vykonání SPARQL dotazu

Aktéři:

- Uživatel.
- Server.

Systém:

- Mobilní aplikace.

Vstupní podmínky:

- Uživatel se nachází na patřičném místě systému.
- Uživatel zná dotazovací jazyk SPARQL.
- Systém má připojení k serveru.

Hlavní scénář:

- Uživatel vytvoří dotaz.
- Uživatel potvrdí dotaz.
- Systém odešle dotaz na server.
- Server zpracuje požadavek a odešle odpověď.
- Systém přijme odpověď
- Systém zpracuje odpověď.
- Systém vizualizuje odpověď.

Alternativní scénáře:

- V případě neplatného dotazu je navraceno chybové hlášení.
- V případě chyby připojení je navraceno chybové hlášení.
- V případě chyby serveru je navraceno chybové hlášení.

Poznámky:

- V případě nenalezení informace je navracena prázdná množina.

2.2.3 Funkční požadavky

Funkčními požadavky (*functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na jednotlivé funkcionality. Pro mobilní aplikaci mezi ně patří především:

1. Aplikace umožní pokročilému uživateli zadat vlastní SPARQL dotaz. Ten se odešle na server, kde se zpracuje, a výsledek se navrátí uživateli.
2. Pro méně znalé uživatele SPARQL a pro uživatele minimalizující použití klávesnice vytvořte předpřipravené dotazy pro základní scénáře použití aplikace.
3. Implementujte funkcionalitu umožňující určení aktuální pozice a její předání dalšímu zpracování.
4. Aplikace dokáže na mapě vizualizovat určitou pozici.
5. Aplikace dokáže vizualizovat HTML data nebo data jako HTML.
6. Aplikace bude poskytovat nápovědu pro snazší použití.
7. Aplikace bude postavená nad daty reprezentovanými výše uvedenou ontologií.

2.2.4 Nefunkční požadavky

Nefunkčními požadavky (*non-functional requirements*) jsou myšleny takové požadavky, které jsou zaměřeny na dílo jako celek, nikoliv na jednotlivé funkcionality. Pro mobilní aplikaci mezi ně patří především:

1. Aplikaci vytvořte v HTML5, aby byla na současných, obzvláště mobilních, zařízeních multiplatformní.
2. Aplikace bude funkční v prohlížečích na nejnovějších verzích OS Android, iOS a Symbian.
3. Aplikace bude funkční v majoritních desktopových prohlížečích.
4. Při vývoji se zaměřte i na použitelnost, aplikaci by měl být schopný, po krátkém zaškolení, používat jakýkoliv běžný student Fakulty informačních technologií.
5. Snažte se nabízet alternativy k zadávání dlouhých textů, aby byla usnadněna obsluha uživatelům mobilních zařízení bez hardwarových klávesnic.
6. Práce nemusí nutně obsáhnout celou zpracovávanou doménu, je ale žádané, aby kvalitně zpracovala hlavní scénáře využití a nebránila dalšímu rozšiřování.
7. Zveřejněte aplikaci a její data pod svobodnými licencemi z rodin GNU, *Creative Commons* a kompatibilními, aby byl umožněn další rozvoj nezávislý na původním autorovi.
8. Aplikace umožní využití off-line. **TODO** Vynechám – muselo by se složitě indexovat.

2.2.5 Návrh architektury

Nasazení mobilní aplikace bylo navrženo tak, jak je vidět na obrázku 2.3. Aplikace komunikuje se SPARQL endpointem serverové aplikace, v případě neposkytování daných informací aplikace komunikuje přímo se zdrojem informací.

2.2.6 Návrh řešení

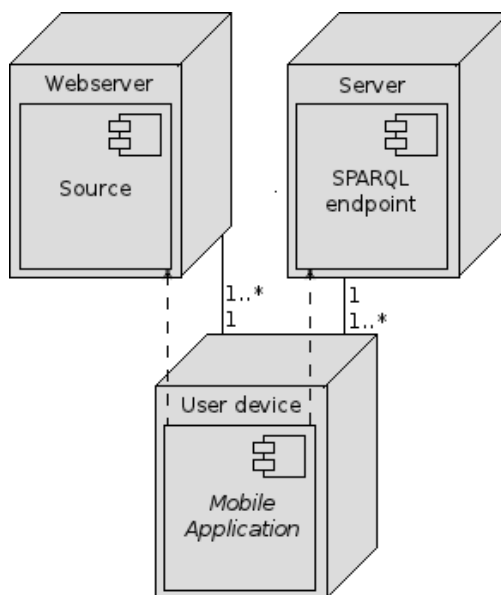
TODO Diskuse alternativ.

TODO Volba implementačního prostředí.

TODO Struktura.

2.2.7 Návrh bezpečnosti

TODO Injekce JS, ošetření vstupů, white-listing URL, elementů, atributů, zakázání cizích zdrojů...



Obrázek 2.3: Diagram nasazení mobilní aplikace

2.2.8 Návrh uživatelského rozhraní

TODO Pojednání o vytvořených prototypch, možná nakonec pod něco zařadím, nebude mít velký rozsah.

2.2.9 Použité technologie

Níže uvádím soupis použitých technologií s odůvodněním, proč byly vybrány a v jakém kontextu použity.

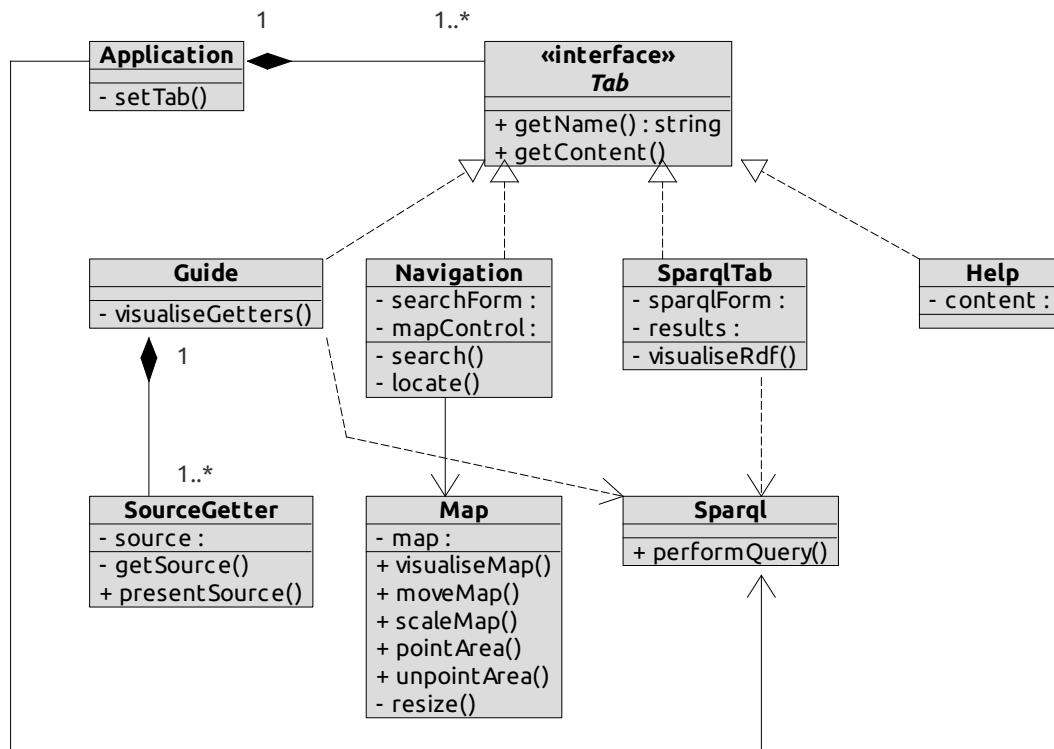
2.2.9.1 HTML5

Většina technologií tvořících mobilní aplikaci by se dala zahrnout do pojmu HTML5 (v širším smyslu významu). Pomocí JavaScriptu se manipuluje s HTML strukturou, která je prezentována za pomoci CSS, klíčová část aplikace – mapa – je tvořena SVG (o tom ještě dále). Technologie byly vybrány pro jejich výbornou portovatelnost – téměř všechna dnešní zařízení je implementují podobně (nebo alespoň mají takové výkonnostní parametry, že na nich lze provozovat přidanou vrstvu abstrahující od skutečné implementace).

2.2.9.2 SVG

Mapa použitá v aplikaci je implementována prostřednictvím SVG. Rozhodnutí bylo učiněno na základě identifikace a porovnání možných technologií:

- Statické obrázky s předgenerovanými proměnnými částmi.



Obrázek 2.4: Třídní diagram mobilní aplikace

- Statické obrázky s dynamicky pozicovanými proměnnými částmi.
- Dynamicky kreslené obrázky na *canvas*.
- Statické SVG s dynamickými úpravami.

První zmíněné řešení bylo zavrženo kvůli nízké flexibilitě – není možné dopředu vygenerovat všechny varianty obrázků, které by pokrývaly možné situace – i kdyby to možné bylo, množství obrázků by bylo velmi vysoké²⁴ a pro omezenou paměťovou kapacitu cílových zařízení nevhodné.

Druhý přístup, dynamické pozicování, byl zavržen z důvodů optimalizací rozvržení prvků mobilními prohlížeči – není zaručené, že se prvek (například ukazatel nalezené místnosti) skutečně umístí, kam má.

Nová vlastnost HTML5, kreslení prostřednictvím JavaScriptu na *canvas* element, byla, narozdíl od předchozích možností, zvažována až do finálního rozhodnutí – ačkoliv se jedná o řešení čistě v režii již používaných technologií (JavaScript, HTML), jeho využití jsem z několika důvodů zavrhl:

²⁴Pro pokrytí všech situací by byly třeba zvlášť obrázky s vyznačením kterékoliv místnosti, kterékoliv bodu zájmu, kterékoliv výchozí pozice... Vzhledem k rozsáhlosti mapy by tak počet obrázků byl enormní.

- Neumožňuje nakreslený obrázek nijak upravit (například posunout ukazatel na mapě), ale musí se vždy překreslit, což je náročnější na zdroje.
- Kvůli principu dynamického generování obrázků JavaScriptem nelze kostru připravit pomocí běžných grafických editorů.
- Subjektivně mi přijde práce s mapou jako SVG obrázkem reprezentovaným XML dokumentem majícím DOM přehlednější, než s méně strukturovatelným JavaScriptem.

Reprezentaci mapy jsem tedy svěřil SVG z následujících důvodů:

- Snadné vytvoření mapy prostřednictvím běžného grafického editoru.
- Možnost znovupoužití mapy standardního grafického formátu i v jiných projektech.
- Práce s mapou (něpř. vyznačování pozic) se provádí pohodou úpravou jejího DOM.
- I po úpravách se stále jedná o týž objekt – není nutné ho překreslovat.

Nutno podotknout, že jsou implementace kreslení na *canvas* a SVG na různých mobilních zařízeních na různých úrovních, ani jedna ovšem nemá výrazně navrch [1] [3].

2.2.9.3 PHP

Z důvodů popsaných v podsekcí 3.2.1.4 (str. 56) bylo nutné vytvořit podpůrnou serverovou aplikaci – proxy. Po úvaze jsem se rozhodl pro využití jazyka PHP, který je podporovaný jako jediný nástroj pro tvorbu dynamického obsahu na fakultním webovém serveru <http://webdev.fit.cvut.cz/>. Tuto kombinaci, vzhledem k její podpoře ze strany fakulty a určení práce pro fakulní nasazení, považuji i přes jinde využitý JavaScript za vhodnou.

2.2.10 Použité knihovny

V práci byla, po úvaze, nakonec využita i práce třetích stran – svobodné knihovny *jQuery* a *HTML Purifier*. Jejich krátký popis a odůvodnění následují:

2.2.10.1 jQuery

A

2.2.10.2 HTML Purifier

Po několika vlastních implementacích nástroje na pročištění a ošetření HTML vstupů pocházejících od třetích stran jsem sáhl k implementaci již prověřené – oblast ošetření aplikace před škodlivými vstupy má z hlediska bezpečnosti vysokou prioritu a neustále se objevující nové hrozby nakonec vedly k tomu, že jsem se rozhodl nespolehat na své vlastní implementace a využít *know-how* tvůrců svobodné PHP knihovny HTML Purifier (<http://htmlpurifier.org/>) – ta odstraní případná nebezpečí za využití pokročilého auditovaného *whitelistu* a v případě potřeby navíc zajistí validitu kódu. Knihovna je využita v rámci proxy zmíněné v předchozím odstavci.

2.2.11 Použité nástroje

Pro tvorbu aplikace byly použity převážně následující nástroje. Majorita jich je uvolněných pod svobodnými licencemi, zpravidla GNU GPL nebo kompatibilními. Jedinou výjimkou jsou některé mobilní prohlížeče, které byly použity k testování aplikace.

2.2.11.1 Kate

Textový editor Kate (<http://kate-editor.org/>) byl, mimo serverové (viz část 2.1.11.1 na straně 42), vybrán i pro tvorbu mobilní aplikace. Nástroj byl zvolen na základě již dříve zmíněných důvodů pro napsání kódu v JavaScriptu, HyperText Markup Language (HTML), Cascading Style Sheets (CSS) i úpravy mapy v Scalable Vector Graphics (SVG).

2.2.11.2 Inkscape

Grafické podklady práce byly vytvořeny v editoru vektorové grafiky Inkscape (<http://www.inkscape.org/>). Nástroj implementuje téměř kompletní specifikaci formátu SVG verze 1.1, je multiplatformní a svobodný. Inkscape byl vybrán kvůli předchozím výborným zkušenostem s prací v něm a celkové vhodnosti pro tvorbu mapových podkladů.

2.2.11.3 Kompare

Nástroj byl zvolen ze stejných důvodů, jak tomu bylo u serverové aplikace (viz část 2.1.11.2 na straně 2.1.11.2).

2.2.11.4 Webové prohlížeče

Pro správné odlazení aplikací, průzkum možností a testování byly vybrány následující prohlížeče. Některé kvůli jejich významnému zastoupení mezi uživateli, jiné pro odlazení aplikace i v jiných implementacích použitých technologií.

- Mozilla Firefox (<http://www.mozilla.org/firefox>).
- Chromium (<http://chromium.org/>).
- rekonq (<http://rekonq.kde.org/>).
- Prohlížeč platformy Android (<http://www.android.com/>) – verze 2 a 4.
- Opera Mobile (<http://www.opera.com/mobile>).
- Internet Explorer Mobile (<http://www.microsoft.com/windowsphone/en-us/features/default.aspx#internet-explorer>). Mimo výše uvedených navíc nabídl pan inženýr Havryluk, vedoucí práce, otestování v následujících prohlížečích:
- Google Chrome for Android (<https://www.google.com/intl/en/chrome/android/>).
- Prohlížeč platformy Android (<http://www.android.com/>) – verze 3.
- Dolphin Browser (<http://dolphin-browser.com/>).
- Nokia Browser for Symbian (<http://browser.nokia.com/>).
- Prohlížeč platformy BlackBerry (<http://us.blackberry.com/smartphones/features/internet.jsp>).
- Safari (<http://www.apple.com/safari/>).

2.2.11.5 Git

Nástroj byl zvolen ze stejných důvodů, jak tomu bylo u serverové aplikace (viz část 2.1.11.6 na straně 2.1.11.6).

Realizace

Během implementace se objevila celá řada zádrhelů, z nichž většina souvisela s doposud ne zcela zdokumentovanými a odlazenými technologiemi, přesto byla velmi přínosná – neustálým hledáním různých řešení a alternativ jsem nutně narazil na mnoho dalších zajímavých postřehů a nápadů. Kapitola se zabývá strukturou aplikací, řešeními problémy, volbou použitých technologií, použitými knihovnami, nádtroji a potenciální aplikací v ideálním světě.

3.1 Serverová aplikace

Aplikace bě

3.1.1 Řešené problémy

TODO Optimalizace, komplikace...

3.1.2 Implementace bezpečnosti

3.2 Mobilní aplikace

Pod pojmem *mobilní aplikace* se skrývá aplikace pevně nezávislá na výše uvedené *serverové*. Jejím hlavním cílem je sloužit studentovi FIT ČVUT jako průvodce, zároveň má ale plnit i účel ukázky využití SPARQL endpointu vytvořeného v *serverové* části.

3.2.1 Řešené problémy

3.2.1.1 Změna velikosti displeje

chrome css v px

3.2.1.2 Chyby prohlížečů v implementaci specifikací

Při práci jsem několikrát narazil na situaci, kdy vše v jednom prohlížeči fungovalo, ve druhém nikoliv, a přesto bylo vše implementované správně podle specifikace. V takovém případě pak bývala chyba na straně prohlížeče, který špatně implementoval určitou specifikaci. Situace nastávala převážně u hodně specifických případů. Příkladem může být chybějící podpora pro nastavení transformace SVG elementu JavaScriptem v prohlížeči Google Chrome [7], tento problém ale lze obejít vlastním sestavením hodnoty celého atributu.

3.2.1.3 Rozdílné DOM u HTML a SVG

SVG je, co se týče návrhu DOM, komplikovanější, než HTML – zatímco u HTML stačí pro přístup k hodnotě atributu

3.2.1.4 Zpracování Cross-Origin zdrojů

3.2.2 Implementace bezpečnosti

TODO Proxy. jQuery...

3.2.3 Ideální implementace

Finální implementace se od té ideální, co se týče omezení vzniklých neideálním světem, moc neliší. Vzhledem k cílení aplikace na současná nová zařízení bylo možné realizovat téměř vše, co bylo naplánováno – v uplynulých letech bylo odvedeno mnoho práce na jednotných standardech napříč zařízeními, a co víc, tyto standardy byly implementovány a z velké části se i dodržují. Problémy se proto vyskytly spíše na poli stále ještě v některých ohledech omezených možnostech mobilních zařízení.

3.2.3.1 Malá velikost displeje

Ačkoliv je toto omezení u mobilních zařízení zřejmé, nic to nemění na faktu, že se s ním musí neustále počítat a je třeba vybírat vhodné komponenty a ty vhodně rozvrhovat. Mým cílem proto bylo vizualizovat jen to nejnutnější.

3.2.3.2 Různé typy polohovacích zařízení

Různá zařízení poskytují různé ovládací prvky využívající různé principy, aplikaci je ale nutné přizpůsobit pro všechny najednou. Jiné je ovládání pro myš (touchpad, trackpad), jiné pro klávesy a jiné pro dotyková zařízení. I přes malý displej je třeba u mobilních dotykových zařízení vytvářet ovládací prvky natolik velké, aby bylo možné je používat prsty.

3.2.3.3 Optimalizace rozvržení prvků

Mobilní zařízení nabízejí pro lepší využití malých displejů mnoho různých optimalizací rozvržení zobrazovaných prvků. Zpravidla jsou tyto vlastnosti velmi užitečné – komprimují nevyužité místo, ve specifických případech, jako je třeba přesné pozicování ukazatele, ale může docházet k obtížně řešitelným situacím. Jediné místo aplikace, které se bez přesného rozmístění neobejde, jsou mapy. Problém jsem se nakonec rozhodl řešit využitím SVG mapových podkladů, ve kterých prohlížeče, zdá se, zatím rozvržení neoptimalizují.

3.2.3.4 Implementace SVG

Implementace SVG je v desktopových prohlížečích na velmi dobré úrovni již nějakou dobu,²⁵ v nedávné minulosti se ale podpora značně zlepšila i na poli mobilních zařízení [1].

U mobilních zařízení nepřekvapuje chybějící implementace některých neklíčových možností SVG, jako je třeba průhlednost v případě Opery Mobile. V této oblasti v současné době rozsáhlému nasazení nepřeje SVG nepodporující hojně využívaný nativní webový prohlížeč Androidu verze 2, od verze 3 je podpora zavedena. Problém lze obejít využitím jiného dostupného prohlížeče.

Překvapením byla pomalá, ačkoliv jinak velmi rozsáhlá, implementace SVG v prohlížeči Mozilla Firefox – po některých operacích, v mém případě to byla třeba změna hodnot atributu `viewBox` SVG elementu (posunutí mapy), prohlížeč vyžaduje překreslení celého obrázku, což je velmi zdlouhavé a zabraňuje to plynulému posunu mapy z aktuální na novou pozici, byť by to bylo vhodné pro lepší uživatelské zorientování se. Tento nedostatek je známý již delší dobu [10].

3.2.3.5 Absence popisků

Velmi užitečnou vlastností dostupnou u desktopových aplikací jsou popisky zobrazující se po najetí myši nad podporovaný prvek. Dá se tímto způsobem velmi dobře implementovat kontextová nápověda, která patří mezi nejpodstatnější zdroje pomoci uživateli. U dotykových mobilních zařízení ale bohužel popisky zpravidla zobrazovat nejdou – najetím, tedy dotykem prstu, se rovnou vyvolá akce prvku, aniž by se popis mohl zobrazit.

3.2.3.6 Dynamické načítání lokálních souborů

Z bezpečnostních důvodů došlo v minulosti v prohlížečích k zamezení načítání JavaScriptových souborů z jiných zdrojů, než ze kterých stránka pochází. To v některých případech²⁶ přerostlo v zamezení dynamického načítání jakýchkoliv

²⁵Posledním významným desktopovým prohlížečem podporujícím SVG se stal s dlouhým odstupem verzí 9.0 Internet Explorer [1], předtím pro něj ale byly dostupné vhodné plugíny.

²⁶Příkladem je Google Chrome.

lokálních souborů – jejich zdroj nemá žádnou hodnotu (`null`), takže je nelze načíst. Aby byla zachována možnost offline využití aplikace na větší skupině zařízení, byla aplikace přepsána na jediné možné řešení – explicitní uvedení všech potenciálně vkládaných souborů, což značně omezilo modularitu aplikace.

3.2.3.7 Otevírání lokálních souborů

Omezení zmíněná v předchozím odstavci stupňuje až do extrémů mobilní prohlížeč Chrome Beta, který nepovolí ani přímé otevření lokálního souboru. **TODO** Ověřit, možná nefunguje celé Chrome...

Testování

4.1 Serverová aplikace

TODO Testování sys., integrace netřeba - běží na jednom místě. Komparativní testování nemá vůči čemu probíhat. Použitelnost netřeba řešit.

4.1.1 Testování programátorem

TODO Kód jsem opětovně revidoval a opravoval chyby. Testování vstupů. Review...

4.1.2 Integrační testování

TODO Po zařazení nové funkcionality byly dotčené části programu otestovány.

4.1.3 Zátěžové testování

4.1.4 Akceptační testování

TODO Bylo prováděno vedoucím práce.

4.2 Mobilní aplikace

4.2.1 Testování programátorem

TODO Kód jsem opětovně revidoval a opravoval chyby.

4.2.2 Testování funkcionalit

TODO Splnění požadavků bylo testováno mnou i vedoucím práce.

4.2.3 Integrační testování

TODO Po zařazení nové funkcionality byly dotčené části programu otestovány.

4.2.4 Testování systémové integrace

TODO Kompatibilita.

4.2.5 Zátěžové testování

TODO Má tu smysl? Proxy? SPARQL?

4.2.6 Akceptační testování

TODO Bylo prováděno vedoucím práce.

4.2.7 Testování použitelnosti

4.2.7.1 Cíl testu

4.2.7.2 Charakteristika účastníků

4.2.7.3 Nastavení testu

4.2.7.4 Seznam testovaných úloh

4.2.7.5 Průběh testu

4.2.7.6 Shrnutí testu

4.2.8 Komparativní testování

TODO Porovnání mobilní aplikace s konkurencí.

Závěr

ZÁVĚR

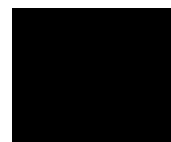
- TODO** Zhodnocení splnění cílů DP.
- TODO** Doporučení pro další pokračování.
- TODO** Shrnutí výsledků a vlastního přínosu.

Literatura

- [1] Deveria, A.: When can I use. . . . 2012,
<http://caniuse.com/>, stav z 1. 5. 2012.
- [2] *Průvodce prváka po ČVUT*. IAESTE ČVUT Praha, 2011.
- [3] Firtman, M.: Mobile HTML5. 2012,
<http://mobilehtml5.org/>, stav z 1. 5. 2012.
- [4] Gayo, J. E. L.: University Ontology. 2011,
<http://purl.org/weso/uni>, stav z 10. 3. 2012.
- [5] Guo, Y.: Lehigh University Benchmark.
<http://swat.cse.lehigh.edu/projects/lubm/>, stav z 1. 5. 2012.
- [6] Hendler, J.; Heflin, J.; Luke, S.: SHOE. 2001,
<http://www.cs.umd.edu/projects/plus/SHOE/>, stav z 1. 5. 2012.
- [7] Laag, L.: SVG transform attribute does not work when manipulated from JavaScript. Září 2010,
<http://code.google.com/p/chromium/issues/detail?id=55010>,
stav z 1. 5. 2012.
- [8] Molnár, J.: *Mobilní navigační systém pro FEL ČVUT*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Květen 2010.
- [9] Styles, R.; Shabir, N.: Academic Institution Internal Structure Ontology (AIISO). 2008,
<http://purl.org/vocab/aiiso/schema>, stav z 1. 5. 2012.
- [10] Bugzilla. 2012,
<https://bugzilla.mozilla.org/>.
- [11] MongoDB. 2012,
<http://www.mongodb.org/>, stav z 5. 5. 2012.
- [12] Node.js. 2012,
<http://nodejs.org/>, stav z 5. 5. 2012.

LITERATURA

- [13] Zemmouchi-Ghomari, L.: Higher Education Reference Ontology (HERO). 2012,
<http://sourceforge.net/projects/heronto/>, stav z 1. 5. 2012.



Seznam použitých zkratek

AIISO Academic Institution Internal Structure Ontology.

BSD Berkeley Software Distribution.

CD Compact Disc.

CSS Cascading Style Sheets.

ČVUT České vysoké učení technické v Praze.

DOM Document Object Model.

DP diplomová práce.

FIT Fakulta informačních technologií.

FOAF Friend of a friend.

GNU GNU's Not Unix!.

GPL General Public License.

HERO Higher Education Reference Ontology.

HTML HyperText Markup Language.

J2ME Java 2 Platform, Micro Edition.

KDE K Desktop Environment.

KIO KDE Input/Output.

Kate KDE Advanced Text Editor.

A. SEZNAM POUŽITÝCH ZKRATEK

LODE Linking Open Descriptions of Events.

LUBM The Lehigh University Benchmark.

MAR měření a regulace.

NTK Národní technická knihovna.

OS operační systém.

OWL-Time An OWL Ontology of Time.

Org An organization ontology.

PHP Hypertext Preprocessor.

RDF Resource Description Framework.

SHOE Simple HTML Ontology Extensions.

SPARQL SPARQL Protocol and RDF Query Language.

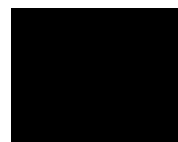
SVG Scalable Vector Graphics.

XML Extensible Markup Language.

cURL Client for URLs.

PŘÍLOHA

B

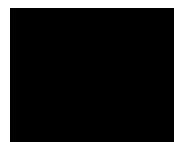


Instalační a uživatelská příručka

TODO spuštění - příkazy

PŘÍLOHA

C



Obsah přiloženého CD

- 12 Bylo prováděno vedoucím práce.
 - 1 Diskuse alternativ.
 - 21 Doporučení pro další pokračování.
 - 4 Injekce JS, ošetření vstupů, white-listing URL, elementů, atributů, zakázání cizích zdrojů...
 - 13 Kód jsem opětovně revidoval a opravoval chyby.
 - 10 Kód jsem opětovně revidoval a opravoval chyby. Testování vstupů. Review...
 - 16 Kompatibilita.
 - 17 Má tu smysl? Proxy? SPARQL?
 - 6 Optimalizace, komplikace...
 - 8 Ověřit, možná nefunguje celé Chrome...
 - 11 Po zařazení nové funkcionality byly dotčené části programu otestovány.
 - 5 Pojednání o vytvořených prototypech, možná nakonec pod něco zařadím, nebude mít velký rozsah.
 - 19 Porovnání mobilní aplikace s konkurencí.
 - 7 Proxy. jQuery...
 - 22 Shrnutí výsledků a vlastního přínosu.
 - 14 Splnění požadavků bylo testováno mnou i vedoucím práce.
 - 23 spuštění - příkazy
 - 3 Struktura.
 - 9 Testování sys,. integrace netřeba - běží na jednom místě. Komparativní testování nemá vůči čemu probíhat. Použitelnost netřeba řešit.
 - 2 Volba implementačního prostředí.
 - 0 Vynechám – muselo by se složitě indexovat.
 - 20 Zhodnocení splnění cílů DP.
- (Shodné položky jsou vynechány.)