

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## INSTANT MESSAGING PRO BLACKBERRY PLAYBOOK

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JIŘÍ HAJNÝ

BRNO 2012



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# **INSTANT MESSAGING PRO BLACKBERRY PLAYBOOK**

INSTANT MESSAGING FOR BLACKBERRY PLAYBOOK

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JIŘÍ HAJNÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**doc. Ing. ADAM HEROUT, Ph.D.**

BRNO 2012

## **Abstrakt**

V práci je představena cílová platforma BlackBerry Tablet OS a její vývojové prostředky. Práce dokumentuje návrh a implementaci klienta komunikačního protokolu XMPP/Jabber. Implementace je provedena ve vývojovém prostředí Flash Builder 4.6, v jazyce Adobe Actionscript 3.0.

## **Abstract**

This thesis introduces the target platform BlackBerry Tablet OS as well as its development tools. It documents design and implementation of a client for communicating protocol XMPP/Jabber named IM Client. The implementation is done in the development environment Flash Builder 4.6 in the programming language Adobe Actionscript 3.0.

## **Klíčová slova**

zasílání rychlých zpráv, BlackBerry PlayBook, XMPP, Jabber, Facebook chat, Google Talk, ActionScript

## **Keywords**

instant messaging, BlackBerry PlayBook, XMPP, Jabber, Facebook chat, Google Talk, ActionScript

## **Citace**

Jiří Hajný: Instant Messaging pro BlackBerry PlayBook, bakalářská práce, Brno, FIT VUT v Brně, 2012

# Instant Messaging pro BlackBerry PlayBook

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Ing. Adama Herouta, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jiří Hajný  
13. května 2012

## Poděkování

Za odborné vedení, cenné rady a ochotu chci poděkovat vedoucímu bakalářské práce doc. Ing. Adamu Heroutovi, Ph.D.

© Jiří Hajný, 2012.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

|  |           |
|--|-----------|
| <b>1 Úvod</b>  | <b>2</b>  |
| <b>2 Platforma BlackBerry Tablet OS</b>  | <b>3</b>  |
| 2.1 Základní pohled . . . . .  | 3         |
| 2.2 Budoucnost . . . . .   | 4         |
| 2.3 Hardware . . . . .   | 4         |
| 2.4 Vývojové nástroje . . . . .  | 4         |
| <b>3 Instant messaging</b>   | <b>8</b>  |
| 3.1 Komunikační protokol . . . . .   | 8         |
| 3.2 IRC . . . . .  | 8         |
| 3.3 ICQ . . . . .  | 9         |
| 3.4 AIM . . . . .  | 9         |
| 3.5 Yahoo! Messenger . . . . .   | 10        |
| 3.6 SKYPE . . . . .  | 10        |
| 3.7 MSN - Windows Live . . . . .   | 11        |
| 3.8 XMPP/Jabber . . . . .  | 12        |
| <b>4 Návrh aplikace IM Client pro BlackBerry PlayBook</b>  | <b>20</b> |
| 4.1 Teoretické poznatky pro návrh uživatelského rozhraní na platformě BlackBerry Tablet OS . . . . . | 20        |
| 4.2 Zvyklosti uživatelského rozhraní pro instant messaging . . . . .                                 | 26        |
| 4.3 Návrh uživatelského rozhraní . . . . .   | 27        |
| 4.4 Návrh struktury aplikace . . . . .   | 32        |
| <b>5 Implementace a testování aplikace</b>   | <b>33</b> |
| 5.1 Struktura aplikace . . . . .   | 33        |
| 5.2 Použití knihoven . . . . .   | 34        |
| 5.3 Problémy s implementací . . . . .  | 34        |
| 5.4 Testování . . . . .  | 35        |
| <b>6 Závěr</b>   | <b>38</b> |
| <b>A Obsah CD</b>  | <b>41</b> |
| <b>B Plakát</b>  | <b>42</b> |

# Kapitola 1

## Úvod

Tablety se v dnešní době dostávají stále více do popředí zájmu. Zastupují v některých aspektech notebooky nebo chytré telefony, akorát se liší velikostí displeje a v možnosti přenosu. Velikostně jsou menší než notebooky, ale výkonnostně se jim přibližují. Jejich displej je použitelnější než u chytrých telefonů např. na prohlížení internetu. Představují jakýsi kompromis mezi přenosností a velikostí displeje. I proto jsem si vybral téma mé bakalářské práce zaměřené právě na jeden z široké nabídky tabletů na trhu, a to BlackBerry PlayBook. BlackBerry PlayBook je prvním, a v současné době jediným, tabletem od kanadské společnosti Research In Motion Ltd. (RIM), výrobce elektronických zařízení BlackBerry. Značka BlackBerry je známa svou kvalitou a vysokým zabezpečením produktů. Bohužel tato platforma není zatím tak rozšířená jako konkurenční platformy tabletů jako např. Apple iOS nebo Android OS (neoficiální informace uvádějí přes 1 milion aktivací PlayBooků). Na druhou stranu má BlackBerry perfektní webové stránky podpory pro vývojáře, spojené s komunitním fórem, kam přispívají i zaměstnanci RIM.

Cílem této práce je vytvořit použitelnou aplikaci pro instant messaging (IM) na této platformě pomocí integrovaného vývojového prostředí Flash Builder, a hlavně popsat tento vývoj. Osobně vlastním tablet BlackBerry PlayBook, který jsem obdržel ve speciální nabídce za nasazení jednoduché aplikace Pexeso na market BlackBerry AppWorld. Stejně jako mnoho dalších lidí i já jsem propadl novému typu moderní komunikace — instant messagingu. Tato práce je dobrou příležitostí nahlédnout „pod kapotu“ IM klientu a jeho tvorbě. Výsledná aplikace bude po společné dohodě s mým vedoucím bakalářské práce klient pro komunikaci pomocí protokolu XMPP s podporou Google Talk a Facebook chatu. V době výběru tohoto protokolu byly na trhu pouze dvě takto zaměřené aplikace, které však nebyly dokonalé. Výrazným motivačním prvkem této práce byl můj osobní rozvoj v objektově orientovaném programování, konkrétně v jazyce Adobe ActionScript 3.0.

V kapitole 2 se budu věnovat popisu platformy BlackBerry Tablet OS, jejích vývojových prostředků a možnostech vývoje pomocí těchto prostředků. Kapitola 3 popisuje několik hlavních služeb IM, podrobně se zabývá protokolem XMPP/Jabber a zdůvodňuje výběr hlavního protokolu výsledné aplikace. Kapitola 4 si klade za cíl objasnit problematiku návrhu aplikace IM Client, shrnuje poznatky pro návrh uživatelského rozhraní aplikace na cílové platformě a zvyklosti uživatelských rozhraní pro IM. Dále je zde uveden návrh struktury výsledné aplikace pro budoucí rozšíření o jiné protokoly. Implementaci a testování je věnována kapitola 5, kde je popsána výsledná struktura aplikace, používání knihoven, problémy s implementací, umístění na market BlackBerry AppWorld a hlavně sběr zpětné vazby a mé reakce na ni. V závěrečné kapitole 6 se nachází zhodnocení celé práce.

## Kapitola 2

# Platforma BlackBerry Tablet OS

V této kapitole je popsána cílová platforma BlackBerry Tablet OS včetně podporovaných zařízení. Kapitola obsahuje seznam dostupných vývojových prostředků a jejich porovnání.

### 2.1 Základní pohled

BlackBerry Tablet OS je operační systém (OS) založený na bázi QNX Neutrino RTOS (Real-time operating system) od firmy QNX Software Systems, která byla akvizicí přijata pod Research In Motion (RIM), výrobce BlackBerry PlayBook. QNX Neutrino je od počátku navržen jako *systém reálného času* a jeho jádro je tzv. *microkernel*. To znamená, že většina operačního systému je spouštěna ve formě malých úloh, které se dají jednoduše vypnout a při chybě neshazují celé jádro. Na rozdíl od OS s velkým komplexním jádrem. QNX Neutrino je OS unixového typu certifikovaný normou POSIX. Je primárně zaměřený na vestavěné systémy, ale uplatňuje se ve více oblastech informační sféry.

BlackBerry Tablet OS podporuje pravý multitasking ve formě systému přepínání mezi běžícími aplikacemi v podobě hlavní nabídky, do níž se lze dostat pomocí gesta tahem z dolní části dotykového okraje (bezel – rámeček) nahoru. Dalšími gesty jsou tahy z levé a pravé části dotykového okraje, které slouží jako rychlé přepínání aplikací bez minimalizace. Gesto shora dolů je aplikačně specifické a nejčastěji se používá pro zobrazení menu aplikace. Pro rychlé zobrazení softwarové klávesnice slouží gesto tažení z levého dolního rohu a pro zobrazení horní systémové lišty tažení z obou horních rohů dovnitř do plochy displeje.

#### 2.1.1 Historie verzí

První vydání byla verze 1.0.1, kterou povýšili na 1.0.3 přidáním možnosti videochatu, úpravou webového prohlížeče a přidáním BlackBerry Bridge, což je program pro spojení tabletu s chytrým telefonem, který ovšem musí mít operační systém BlackBerry OS 5.0 a vyšší. Spojení umožňuje ovládání tabletu přes telefon a ovládání některých programů telefonu (např. kontakty, email, kalendář a souborový manažer) z tabletu. Další velkou aktualizací se stala verze 1.0.7, která už podporuje mnoho jazykových mutací, zlepšuje Wi-Fi konektivitu a BlackBerry Bridge a podporuje aplikace vytvořené ve vývojovém prostředí Native SDK (viz dále). Touto dobou se BlackBerry Tablet OS vyskytuje ve verzi 2.0.1.358, která již obsahuje nativního klienta pro zprávy (email, Facebook), kontakty a kalendář, dále podporuje spouštění aplikací ze systému Android (musí být naportované – viz dále), integruje sociální síť jako Facebook, Twitter a LinkedIn a vylepšuje aplikace jako VideoChat, DocsToGo a další.

## 2.2 Budoucnost

Společnost RIM do budoucna plánuje sjednocení operačních systémů pro chytré telefony a pro tablety za vzniku jednotné platformy BlackBerry 10 (BB10), též založené na platformě QNX. Tato platforma je již ve stádiu vývoje, důkazem toho je uvolnění vývojových nástrojů na konferenci BlackBerry World 2012 ve dnech 1.-3.5.2012, kde každý návštěvník konference dostane limitovanou edici vývojářského prototypu – BlackBerry 10 Dev Alpha. RIM se chce v budoucnu převážně zaměřit na HTML5 a Flash. Čerpáno z [10].

## 2.3 Hardware

Do dnešního dne bylo vydáno pouze jedno zařízení pro platformu BlackBerry Tablet OS a to dne 19. dubna 2011 BlackBerry PlayBook, také od společnosti RIM. Jedná se o poměrně nový elektronický výrobek. Obsahuje integrovaný Adobe Flash 11.2 a podporuje HTML5. V tabulce 2.1 jsou uvedeny základní technické specifikace. Jeho rozměry činí 194mm x 130mm x 10mm. Váží 425g. Existuje ve třech verzích rozlišených podle velikosti vestavěné paměti – 16GB, 32GB a 64GB. Na snímání obrazu je vybaven dvěma kamerami, přední 3MP a zadní 5MP, obě schopné natáčet HD video (1080p). Pro přehrávání audia slouží stereo reproduktory. Tablet nemá zabudován vibrační modul. Vpředu je vestavěn snímač okolního osvětlení a LED dioda k upozornění na nově příchozí zprávu. Dotykový displej je schopen určit až 4 dotyky současně [11].

|               |   |
|---------------|---|
| Displej       | 7" (17,8cm) LCD 1024x600 17:10 Multi-dotykový kapacitní   |
| Procesor      | 1 GHz Dual-core   |
| Paměť         | 1GB RAM, 16-64GB Flash  |
| Příslušenství | Akcelerometr, 6-osý pohybový sensor (gyroskop), digitální kompas (magnetometr) a GPS                      |
| Konektivita   | Wi-Fi 802.11 a/b/g/n, Bluetooth v2.1 + EDR<br>microUSB, microHDMI, 3-pinový konektor pro dokovací stanici |

Tabulka 2.1: Základní specifikace tabletu BlackBerry PlayBook

## 2.4 Vývojové nástroje

Pro vývoj na platformě BlackBerry Tablet OS existuje hned několik vývojových prostředí, každé určeno pro jiný jazyk a jiný přístup. Pro nativní jazyk C/C++ se používá prostředí *BlackBerry Native SDK for Tablet OS*. V jazyce HTML5 a Javascriptu se programuje v prostředí *BlackBerry WebWorks for Tablet OS*. Dále, jak již bylo uvedeno, lze naportovat aplikace napsané na Android OS v prostředí *Android Runtime*. A konečně hlavní část mé bakalářské práce zahrnuje vývoj v jazyce Adobe ActionScript 3.0 v prostředí Adobe AIR - *BlackBerry Tablet OS SDK*. Ke každému prostředí je dostupný i simulátor. Vše je zcela zdarma.

### 2.4.1 Native SDK

BlackBerry Native SDK for Tablet OS zahrnuje všechny nástroje, které jsou potřeba k vývoji aplikací včetně překladače, linkeru, knihoven, vývojového prostředí a nástrojů příkazové



řádky. Vývojové prostředí se zde jmenuje QNX Momentics IDE a je založeno na základu vývojového prostředí Eclipse. Testovat je možno buď přímo na tabletu BlackBerry PlayBook nebo na simulátoru, který se dá volitelně stáhnout. Pokud chcete nativní aplikaci zkoušet na svém tabletu, musí mít verzi software minimálně 1.0.7. Na oficiálních stránkách [12] je dostupné ke stažení velké množství příkladů a vzorových aplikací pro rozjezd vývoje bez jakýchkoliv předešlých znalostí.

#### 2.4.2 WebWorks SDK

WebWorks umožňuje vývojářům vyvíjet webové aplikace pro BlackBerry chytré telefony a tablety pomocí HTML5, CSS a Javascriptu s plným rozsahem využití BlackBerry API a služeb. Funkce v API umožňují například zaslání sms z aplikace, používat protokoly hovorů, sledování, zda je uživatel momentálně na telefonu, přístup ke zmeškaným souborům a také ovládání přehrávání a navigace mezi audio soubory. WebWorks obsahují několik přístupových bodů, které umožňují vývojářům spouštět aplikace v pozadí a z jakéhokoli menu. Aplikace mohou také běžet na pozadí, zatímco si uloží obsah webové stránky, což má příznivý vliv na výkon a paměť.[8]Na stránkách [13] lze volitelně stáhnout simulátor Ripple (což je multiplatformní emulátor mobilního prostředí), běžný simulátor PlayBooku a celé SDK, které zahrnuje nástroje pro kompilaci, balení a nasazení do marketu.

#### 2.4.3 Android Runtime

Zaměstnanci RIM vyvinuli několik způsobů jak naportovat aplikaci ze systému Android OS na BlackBerry Tablet OS. Lze stáhnout plugin do IDE Eclipse, který umožňuje aplikaci ověřit, přebalíkovat a podepsat (viz dále). Nebo je možno použít online webové rozhraní, které požaduje jen nainstalované Android SDK. Poslední možností je využití nástrojů příkazové řádky. Portování ale nelze uplatnit na všechny aplikace kvůli integraci např. Google Maps nebo jiných Google produktů. Podporována je zatím pouze verze Androidu 2.3.3.

#### 2.4.4 BlackBerry Tablet OS SDK

Pro použití BlackBerry Tablet OS SDK je potřeba stáhnout Adobe Flash Builder 4.6 a do něho pak nainstalovat samotné SDK. Programovacím jazykem je zde buď jazyk *Adobe ActionScript 3.0* nebo jazyk *MXML* z Adobe Flex, což je open source aplikační framework, který umožňuje jednoduše vytvářet mobilní aplikace pro iOS, Android a BlackBerry Tablet OS zařízení, stejně jako tradiční aplikace v prohlížeči a na desktopu využívajíc přitom stejný programovací model, nástroje a knihovny.

Můj výběr jazyka ActionScript byl dán tím, že preferuji objektové programování a v době výběru ještě nebyla možnost portování aplikací psaných na Android. V jazyce ActionScript se dají vytvořit stejné programy jako ve frameworku Flex s tím rozdílem, že ve Flexu je kód kratší ale méně přehledný.

ActionScript je objektový vysokoúrovňový jazyk. Pomocí ActionScriptu se dají vytvářet internetové aplikace, animace i desktopové aplikace. ActionScript je úzce spjatý s platformou Flash. Vyvíjely se společně, protože platforma Flash vznikla roku 1996. V dnešní době představuje kombinace návrhových a animačních nástrojů prostředí Flash CS5 a pokročilých schopností jazyka ActionScript 3.0 jedno z nejmocnějších, nejuniverzálnějších a nejoblíbenějších vývojových prostředí. Jazyk ActionScript však začínal jako součást platformy Flash poměrně skromně.

## Stručná historie platformy Flash a jazyka ActionScript

V prvních třech verzích platformy Flash nebyl k dispozici žádný programovací nástroj. Platforma Flash 4 byla první verzí, která umožňovala vkládat zdrojový kód pomocí jednoduchého skriptovacího jazyka, jenž se začal neformálně nazývat jazyk ActionScript. U platformy Flash 5 se jazyk ActionScript více vyvinul a stal se z něj oficiální skriptovací jazyk. S každou další verzí platformy Flash schopnosti jazyka ActionScript rostly a tento jazyk začal nabízet interaktivní řízení animací, textu, zvuků, videa, dat atp. V roce 2003 vznikl jazyk ActionScript 2.0 a jeho dovednosti byly srovnatelné s objektově orientovanými programovacími jazyky, jako jsou jazyky Java nebo C#. Přestože tento jazyk dostačoval ostatním programovacím jazykům v nabídce funkcí, nestačil na ně efektivitou. V roce 2006 představila společnost Adobe jazyk ActionScript 3.0, jenž nabízel nové funkce a rovněž dramatický nárůst efektivity. Prostředí Flash CS4 přidalo k jazyku ActionScript 3.0 další funkce – včetně práce s 3D grafikou, novým řízením animací a třídy jazyka ActionScript pro práci s platformou Adobe AIR. Prostředí Flash CS5 pokračuje ve vývoji jazyka ActionScript 3.0 a přináší spoustu nových pokročilých funkcí – například vylepšování platformy Adobe AIR, práci s různými zařízeními a radiči, a to včetně vícedotykových a běžných dotykových zařízení.

### Nevýhody oproti starším verzím jazyka ActionScript

- Je nutné psát více zdrojového kódu, aby bylo dosaženo stejných výsledků.
- Flashové aplikace napsané v jazyce ActionScript 3.0 nelze také začlenit do projektů vytvořených s dřívějšími verzemi tohoto jazyka, protože přehrávač Flash Player 9 a novější ve skutečnosti obsahuje dva přehrávače jazyka ActionScript. Přestože spolu můžou komunikovat soubory mezi dvěma virtuálními stroji, není to tak jednoduché jako komunikace mezi soubory určenými pro stejnou verzi virtuálního stroje AVM (ActionScript Virtual Machine).

### Výhody oproti starším verzím jazyka ActionScript

- Lepší efektivita. Jak už bylo zmíněno, kód jazyka ActionScript 3.0 se provádí mnohem rychleji než kód starších verzí tohoto jazyka – obvykle dvakrát až desetkrát rychleji, ale v některých případech se může provést až stokrát rychleji.
- Konzistentnější syntaxe. Protože jazyk ActionScript stavěl až do verze 2.0 na starších verzích jazyka, často existovalo více způsobů, jak udělat stejný úkon. To se mohlo zdát velmi nejasné. Ve starších verzích jazyka ActionScript se například mohly podstatně lišit tak jednoduché věci, jako je odpovídání na události nebo vytváření nového objektu, a to v závislosti na typu události nebo objektu. V jazyce ActionScript 3.0 například existuje jen jediný způsob, jak naslouchat událostem a jak na ně odpovídat, a to bez ohledu na typ událostí.
- Lepší kontrola chyb a zpětná odezva.
- Spousta nových funkcí. Jazyk ActionScript 3.0 zavedl řadu nových tříd, které nabízejí dříve nedostupnou funkčnost, včetně funkcí pro práci se zvukem, videem, textem, formátem XML, 3D grafikou a tak dále.

## Prostředí Flash CS5, Flash Builder 4 a Flex

Velké množství uživatelů prostředí Flash zná prostředí Adobe Flash Builder a Flex, ale neví, jestli a jak je využít ve svém vývojovém procesu. Prostředí Flash CS5 a Flash Builder jsou komerční aplikace od společnosti Adobe. Flash Builder je nový název prostředí, které bylo dříve známé jako Flex Builder. V prostředí Flash CS5 nebo Flash Builder lze vytvářet soubory SWF pro přehrávač Flash Player i samostatné aplikace platformy Adobe AIR. Další možnost pro zkušené programátory je použít bezplatnou sadu Flex SDK, již dala k dispozici společnost Adobe na svých internetových stránkách. Všechny uvedené programy podporují celý jazyk ActionScript 3.0. Prostředí Flash Builder se více zaměřuje na pokročilejší programátory a obsahuje několik funkcí, které lze použít při vývoji rozsáhlých RIA aplikací a projektů řízených daty. Na druhou stranu – prostředí Flash CS5 obsahuje nástroje a rozhraní přizpůsobené potřebám návrhářů a tvůrců animací.

Čerpáno z [3].

### 2.4.5 Shrnutí

Pro všechny možnosti programování na BlackBerry Tablet OS jsou ke stažení všemožné příklady a vzorové aplikace pro urychlení vývoje v daných prostředích. Po vytvoření aplikace je nutné aplikaci podepsat. K tomu je zapotřebí tzv. *signing keys* – podepisovací klíče. O ně je potřeba si zažádat na speciálních internetových stránkách. Po zažádání jsou odeslány na email žadatele dva „.csj“ soubory, RDK a PBDT. První je potřeba zaregistrovat u RIM, což je možné udělat za pomoci jednoduchého průvodce ve Flash Builderu. Druhý slouží pro vytvoření tzv. *debug tokenů*, což jsou soubory, které je potřeba nahrát na PlayBook, aby bylo možné aplikaci ladit. Jsou vázány na specifického autora a konkrétní tablet určený svým PINem. Tímto mechanismem je celkem dobře zabezpečena původnost aplikací a tím i jejich bezpečnost.

## Kapitola 3

# Instant messaging

Tato kapitola obsahuje základní popis některých hlavních komunikačních služeb a jejich protokolů a zaměřuje se podrobně na protokol XMPP/Jabber. Na začátku každé podkapitoly o IM protokolu je malá tabulka s výčtem několika vlastností, podle kterých jsem se mimo jiné rozhodoval o výběru IM protokolu pro moji aplikaci.

Instant messaging (IM) znamená v překladu zasílání rychlých zpráv. Je to internetová služba, umožňující svým uživatelům sledovat, kteří jejich přátelé jsou právě připojeni, a dle potřeby jim posílat zprávy, chatovat, přeposílat soubory mezi uživateli a i jinak komunikovat. Hlavní výhodou oproti např. e-mailu je, že zprávy jsou doručeny v reálném čase a že druhá strana ví, zda je účastník k dispozici, či nikoliv. První IM službou byla síť IRC (Internet Relay Chat), která vznikla už v roce 1988. Během několika let bylo vyvinuto mnoho alternativních IM klientů, každý s vlastním protokolem (navzájem nekompatibilním). Rozdíly mezi protokoly stírá otevřený XMPP protokol pro IM, který dokáže mimo jiné pomocí takzvaných transportů komunikovat i s ostatními sítěmi.

### 3.1 Komunikační protokol

Pokud spolu chtějí po síti komunikovat dva určité koncové body, je především zapotřebí, aby byla jasně dána pravidla jejich komunikace, čímž se zabezpečí, že si budou oba subjekty rozumět. Soubor takových pravidel se nazývá standard neboli protokol. Protokol může mít formální popis a obsahuje popis syntaxe, sémantiky a synchronizace komunikace.

### 3.2 IRC

| Licence | Centralizace     | Open source knihovny           |
|---------|------------------|--------------------------------|
| Open    | decentralizováno | ANO – AS3IRCLib, flexircclient |

IRC je zkratka pro Internet Relay Chat. IRC protokol je podrobně popsán v dokumentu RFC (Request For Comments) 1459 [2] a je nadstavbou protokolu TCP. Zajišťuje tyto služby:

- Doručování zpráv mezi serverem a klientem.
- Lokalizace jednotlivých klientů - server uchovává po celou dobu připojení klienta informaci o jeho umístění v síti.
- Správa kanálů.

Síť je založena na modelu klient-server. Komunikace probíhá na portech 6660-6669. Jakmile se klient se serverem spojí, obdrží unikátní devítimístné jméno, které si server uchovává. Uživatelé sítě spolu komunikují ve virtuálních místnostech – v tzv. *kanálech*. Název kanálu vždy začíná znakem # (popř. &) a nesmí obsahovat mezery, čárky a být delší než-li 200 znaků. Počet kanálů, ve kterých může uživatel současně být, je neomezen a současně není omezen ani počet konverzací. V kanálu je možné vést buď hromadnou konverzaci nebo soukromý hovor. Založit nový kanál může kdokoli z uživatelů, tím se automaticky stává správcem tohoto kanálu a jsou mu přiřazeny nejvyšší práva, těmi jsou například: vyhození uživatele z kanálu (popř. blokáce přístupu přezdívkou či IP adresy), změna režimu kanálu (volný vstup, vstup chráněný heslem, vstup pouze na pozvání). Správce kanálu může zvolit další uživatele, kteří mu se správou budou pomáhat. Ti už mají jen částečná privilegia. Takový uživatel se nazývá *channell operator* a má znak @ umístěný vedle své přezdívky.

### 3.3 ICQ

| Licence      | Centralizace                     | Open source knihovny |
|--------------|----------------------------------|----------------------|
| Proprietární | centrální server – login.icq.com | NE                   |

ICQ je komunikační služba, pomocí které je možné mezi uživateli posílat nejen textové zprávy, ale i soubory, nebo hrát jednoduché hry a další funkce přidané pomocí zásuvných modulů (pluginů). Byla vyvinuta Izraelskou firmou Mirabilis v roce 1996. Později, 8. června 1998, ICQ koupila americká firma AOL (America Online) za 407 miliónů dolarů. Nynějším vlastníkem je firma Digital Sky Technologies, která ICQ koupila v dubnu 2010 od firmy AOL za 187,5 miliónu dolarů.

Každý uživatel ICQ má přiděleno tzv. *identifikační číslo UIN* (Universal Internet Number nebo Unified Identification Number). Toto číslo obdrží každý uživatel ICQ ihned po registraci svého účtu. Je to jediná identifikace uživatele, která se nedá u již registrovaného uživatele změnit. Všechny ostatní identifikační údaje, jako je jméno, přezdívka, e-mail a další, jsou modifikovatelné.

ICQ využívá pro svou komunikaci protokol OSCAR (Open System for CommunicAtion in Realtime). Dříve bylo možné používat protokol TOC, pro který 19. srpna 2005 AOL ukončila podporu a ke konci roku 2007 jej vypnula. Jakmile byl TOC protokol vypnut, přestali neoficiální klienti využívající protokolu TOC fungovat. V Česku zajišťovala do 13. března 2012 podporu, oficiální překlad a prezentaci klienta ICQ firma Centrum Holdings, provozovatel portálů Centrum.cz a Atlas.cz. Od tohoto data bude zajišťovat tuto podporu provozovatel TV Nova - společnost CET 21. ICQ má tu výhodu, že je v České republice ještě stále poměrně rozšířený, avšak má i dost nevýhod. Asi tou nejdůležitější nevýhodou je uzavřenost služby a protokolu společně s velmi diskutovanou licencí.

### 3.4 AIM

| Licence      | Centralizace     | Open source knihovny |
|--------------|------------------|----------------------|
| Proprietární | centrální server | ANO – částečné       |

Název je zkratka slov AOL Instant Messenger (AOL - America OnLine). Síť AIM je používána především v USA. Oficiální klient podporuje OS Windows, Mac OS, iOS, Android, BlackBerry OS a Windows Mobile. Jeho počátky sahají do roku 1994. Nejprve byl komunikátor užíván jen vnitropodnikově, od května 1997 je k dispozici všem. AIM se postupně

vyvíjel a ze systému, který původně umožňoval jen sledovat stav kontaktu a zaslání IM zpráv se stal systém podporující také videokonference. Poslední verze klienta AIM 1.0.1.2 podporuje textové zprávy, voice chat, video konference, emaily a doplňky jako jsou přenos souborů, adresář, přeposílání zpráv i na mobilní telefon, spolupráci se sociální sítí Facebook či Twitter. Standardní protokol, který je využit v AIM je tentýž jako v systému ICQ – OSCAR. Firma AOL vytvořila také jednodušší protokol TOC, kterému chybí hodně vlastností z protokolu OSCAR, ale zajistí základní chatovací funkce. Specifikace protokolů TOC/TOC2 byly uvolněny firmou AOL, zatímco protokol OSCAR je uzavřený a musel být podroben reverznímu inženýrství. Uživatelé AIM jsou identifikováni dle tzv. *screen name* či emailové adresy ve tvaru *uzivatel@aim.com*. Screen name je řetězec písmen či číslic začínající písmenem. AIM podporuje tzv. *Real-Time IM*, což znamená, že uživatel vidí přímo co a jak rychle druhý uživatel píše.

### 3.5 Yahoo! Messenger

| Licence      | Centralizace                | Open source knihovny        |
|--------------|-----------------------------|-----------------------------|
| Proprietární | centrální servery (asi 100) | ANO – nedodělané, nefungují |

Yahoo! Messenger je další z řady zástupců IM klientů podporující výměnu textových zpráv, video přenos, chat, sdílení fotek, volání a další vlastnosti. Tato služba je spjata s americkým portálem Yahoo.com. Oficiální klient je poskytován pro operační systémy Windows, Linux a MacOS, protokol podporují také další komunikační programy jako Gaim, Miranda IM a Trillian. Pro využívání je nutné zaregistrovat tzv. *Yahoo! ID*. Yahoo messenger používá protokol YMSG (Yahoo! Messenger Protocol), který je proprietární, ale dají se na něj sehnat knihovny pro jazyky Java, php, ActionScript a další.

### 3.6 SKYPE

| Licence      | Centralizace                               | Open source knihovny |
|--------------|--|----------------------|
| Proprietární | centrální servery na přihlášení, jinak P2P | NE                   |

Skype je pravděpodobně jediný IM klient na bázi *peer-to-peer* (P2P), to znamená, že je decentralizovaný a na svůj běh používá počítače zapojené do sítě Skype. Chat je v tomto IM klientu až na druhém místě po volání přes internet (VoIP). Jeho autory jsou Estonci Niklas Zennström a Janus Friis, tvůrci populárního softwaru Kazaa. Program umožňuje telefonovat mezi svými uživateli zdarma, za poplatek lze telefonovat do tradičních telefonních sítí (služba SkypeOut) a případně získat telefonní číslo a přijímat telefonáty z pevných a mobilních sítí se službou SkypeIn. Skype však nepodporuje volání linek záchranného systému (rychlá lékařská pomoc, hasiči, policie). Komunikace je šifrována šifrou AES o délce klíče 256 bitů. Komunikační protokol ani zdrojové kódy programu nejsou veřejně dostupné. Po vědomí o fungování protokolu je díky reverznímu inženýrství. K prolomení došlo v červenci 2006 týmem čínských inženýrů.

### 3.7 MSN - Windows Live

| Licence      | Centralizace      | Open source knihovny         |
|--------------|-------------------|------------------------------|
| Proprietární | centrální servery | ANO – nekompletní, zastaralé |

Zkratka MSN znamená Microsoft Network a označuje celou řadu internetových služeb od společnosti Microsoft. Původně bylo MSN spuštěno 24.8.1995 při příležitosti vydání Windows 95. V roce 2006 byla služba přejmenována na Windows Live kvůli snaze zvýšit atraktivitu poskytovaných služeb pro širší masu obyvatel. Ve Spojených státech amerických není MSN jen internetovou službou, ale i poskytovatelem internetu. Pokud chcete užívat služby MSN musíte mít *Windows Live ID*, které je možno získat zdarma a dovoluje přístup ke všem službám. Je vázáno na emailovou adresu, kterou vyplníte při registraci. Pro použití je nunto mít nainstalováno Windows Live Essentials, které v sobě zahrnuje program pro instant messaging Windows Live Messenger (WLM). Samotný klient používá protokol *Microsoft Notification Protocol* (MSNP, také známý jako Mobile Status Notification Protocol), který je nyní ve verzi 19. Čerpáno z [9]. Každý měsíc používá WLM více než 300 miliónů lidí v 76 zemích a 48 jazycích [7].

## 3.8 XMPP/Jabber

| Licence | Centralizace     | Open source knihovny         |
|---------|------------------|------------------------------|
| Open    | decentralizováno | ANO – XIFF, seesmic-as3-xmpp |

Název je odvozen zkrácením *eXtensible Messaging and Presence Protocol*, neboli „rozšiřitelný protokol pro posílání zpráv a zjištění stavu“. Původně vznikl jako protokol pro IM síť Jabber. Brzy se ale ukázalo, že kromě IM může být výhodně využito i pro vzájemnou komunikaci programů nebo pro ovládání různých automatických služeb (botů). Později byl začleněn jakožto standard Internetu do RFC dokumentů – základní normy jsou RFC 3920 (obecná specifikace protokolu) a RFC 3921 (samotný instant messaging a zobrazení stavu), které byly ale v březnu roku 2011 revidovány a vznikly novější verze RFC 6120<sup>1</sup>, RFC 6121<sup>2</sup> a RFC 6122<sup>3</sup>. RFC obsahující některá další rozšíření XMPP protokolu jsou například RFC 3922 a RFC 3923. O vývoj protokolu se stará XMPP Standards Foundation (XSF)<sup>4</sup>. Rozšíření nad rámec RFC jsou vydávána v podobě tzv. XEP (XMPP Extension Protocol). XMPP je implementací obecného značkovacího jazyka XML (*eXtensible Markup Language* — viz dále). Specifikace jsou zcela otevřené a dostupné všem, kdo chtějí implementovat software s podporou XMPP. Servery XMPP protokolu používají standardně TCP port 5222, nebo port 5223 pro komunikaci šifrovanou SSL<sup>5</sup>/TLS<sup>6</sup>. Pro vzájemnou komunikaci serverů je pak vyhrazen port 5269 [5].

Pro provoz IM je nutné mít registrováno tzv. *Jabber ID* (JID), které slouží stejně jako emailová adresa pro email. Forma celého JID je ve tvaru `uzivatel@host/zdroj`, kde `uzivatel` slouží k přihlášení (login) do sítě a `host` je názvem serveru, kde je uživatel registrován. Jako kontaktní informace se používá jen tzv. *holé JID* (bare JID), tzn. bez zdroje. *Zdroj* (anglicky resource) je pak řetězec identifikující zařízení/platformu, ze kterého se uživatel přihlásil. Buď je možno si ho zvolit, nebo je přidělen serverem. Může být více různých zdrojů přihlášených současně a priorita určuje, kam se má zasílat nově příchozí zpráva (viz dále). Seznam kontaktů se uchovává v tzv. *rosteru*, který je uložen na serveru a při každé změně se synchronizuje se všemi připojenými klienty. Kontakty je možno rozdělit do skupin podle uživatelových pravidel.[6]

### 3.8.1 Kritéria výběru protokolu

Výběr tohoto protokolu jako hlavního pro moji aplikaci byl způsoben tím, že – jak vypovídá z malých tabulek na začátcích každé podkapitoly o IM protokolech – je jeden z mála otevřený, decentralizovaný a jsou implementovány funkční (téměř, jen s malými úpravami) opensource knihovny. Dalšími kritérii byly:

- Počet aplikací pro daný protokol na dané platformě.
- Užití daného protokolu v nejvíce používaných službách (Facebook, Google Talk).

V době vybírání protokolu byla jen jedna aplikace pro protokol Jabber, a to ještě dost nekvalitně zpracovaná. Výhoda v podobě rozšíření protokolu v hodně využívaných službách jako jsou Facebook či Google Talk byla také rozhodující.

<sup>1</sup><http://tools.ietf.org/html/rfc6120>

<sup>2</sup><http://tools.ietf.org/html/rfc6121>

<sup>3</sup><http://tools.ietf.org/html/rfc6122>

<sup>4</sup>[www.xmpp.org](http://www.xmpp.org)

<sup>5</sup>Secure Sockets Layer

<sup>6</sup>Transport Layer Security



Dalším kandidátem na hlavní protokol by býval mohl být protokol IRC, jelikož má stejné „tabulkové“ atributy (otevřený, decentralizovaný a funkční opensource knihovny). Protože však k protokolu IRC již na začátku mé práce bylo na dané platformě velké množství funkčních aplikací, rozhodl jsem se od tohoto protokolu upustit. Naopak od většiny ostatních protokolů nebyli v době výběru žádní klienti a tudíž jsem se dále rozhodoval dle funkčnosti a schopnosti opensource knihoven. Kupříkladu pro protokol OSCAR ze služby ICQ neexistuje žádná opensource knihovna a její vytvoření by bylo časově velice náročné, přičemž by se mohla specifikace protokolu rázem změnit — díky uzavřenosti protokolu by se to projevilo jen nefunkčností knihovny — a veškerá práce by neměla užitek. Tomu jsem chtěl zabránit vybráním standardu jako hlavního protokolu.

### 3.8.2 Historie

Technologii Jabber/XMPP vyvinul Jeremie Miller v roce 1998. Jeremiho už nebavilo používat čtyři různé klienty pro uzavřené IM služby a rozhodl se vytvořit open source server jabberd a zveřejnil ho 4. ledna roku 1999. Dlouho předtím mu komunita vývojářů pomáhala a vytvořila open source klienty pro Linux, Macintosh a Windows, a také knihovny pro jazyky Perl a Java. Během let 1999 a 2000 se komunita zaměřila na detaily protokolu XMPP a nakonec v květnu roku 2000 zveřejnili jabberd ve verzi 1.0. V srpnu roku 2001 byla zformována organizace Jabber Software Foundation, která byla přejmenována v roce 2007 na XMPP Standards Foundation. V říjnu roku 2004 byly vydány standardizace protokolu komisí IETF (Internet Engineering Task Force) ve formě RFC a to konkrétně [RFC 3920] a [RFC 3921]. Tato komise mimo jiné standardizovala většinu technologií pro fungování internetu (včetně TCP/IP, HTTP, SMTP, POP, IMAP a SSL/TLS). Publikací těchto RFC došlo k širokému rozšíření technologií XMPP. V srpnu roku 2005 byly na bázi XMPP spuštěny služby Google Talk IM a VoIP (Voice over Internet Protocol). Stovky dalších služeb následovaly. Velké a úspěšné softwarové firmy, jako např. Apple, Cisco, IBM, Nokia a Sun, využívají XMPP v jejich produktech. Nespočet firem, universit a vlád používá své systémy IM založené na XMPP. Čerpáno z [6]. V březnu roku 2011 byl protokol revidován a byla sepsána nová RFC.

### 3.8.3 XML

Veškerá komunikace mezi koncovými body probíhá pomocí výměny dat, která jsou ve formátu XML. XML je odlehčenou verzí standardu SGML (Standard Generalized Markup Language). Tvůrcem specifikace je W3C (World Wide Web Consortium), která vydala doporučení XML ve verzi 1.0 v únoru 1998. Specifikace obsahuje jak syntaktické požadavky jazyka, tak i požadavky na parsery. Každý XML dokument je složen ze značek (*tagů*) – *párových* (mají otevírací `<x>` i uzavírací `</x>` značku) a/nebo *nepárových* (pouze jedna značka `<x/>`). Každý dokument taktéž musí obsahovat právě jeden kořenový prvek, který tvoří „obálku“ pro obsah dokumentu. Značky do sebe mohou být vnořeny, ale nesmí se křížit – každý nekořenový prvek musí být uzavřen do jiného prvku.

Z hlediska zpracování rozlišujeme 2 typy dokumentů – *správně sestavené* a *validní*. Správně sestavený dokument vyhovuje všem syntaktickým pravidlům stanoveným ve specifikaci. Nesprávně sestavené dokumenty musí být parserem odmítnuty. Validní dokument je správně sestavený dokument, který splňuje další sémantická pravidla, která mohou být stanovena například v definici typu dokumentu (DTD) nebo XML schématu (XML Schema). Dokument tedy může být správně sestavený, ale když bude obsahovat neznámou značku (z pohledu DTD nebo XML schématu), není validní. Převzato z [1]

XMPP je streamovací protokol. To znamená, že zprávy (části XML) jsou doručovány v obou směrech, přičemž celá interakce je zabalena do jednoho XML dokumentu ve směru klient <-> server a jednoho dokumentu v opačném směru. [5] Struktura XML dokumentu v XMPP je definována pomocí XSD schémat <sup>7</sup>. Schémata jasně udávají, jak má XML dokument pro komunikaci vypadat. Změnou hodnoty jednotlivých XML elementů se změní druh požadavku či odpovědi na požadavek.

### 3.8.4 Jádru XMPP

XMPP technologie podporuje jen tři základní XML elementy pojmenované *XML stanzy*:

- `<iq/>` – nositel informačních a dotazových zpráv
- `<presence/>` – nositel zpráv o dostupnosti a přítomnosti
- `<message/>` – nositel běžných konverzačních prvků

Celý tok těchto XML stanz je zahrnut do tzv. *XML Streamu*, který představuje celkovou komunikaci klient <-> server. Uzavírá se do párového tagu `<stream>` a `</stream>`. V praxi se zapisuje jako `<stream:stream>`, což znamená, že je to tag `stream` definován a kvalifikován jmenným prostorem `stream`, který je definován na URI: <http://etherx.jabber.org/streams>. Ukončení streamu se pak způsobí zasláním tagu `</stream:stream>`.

#### Prvek `<iq>`

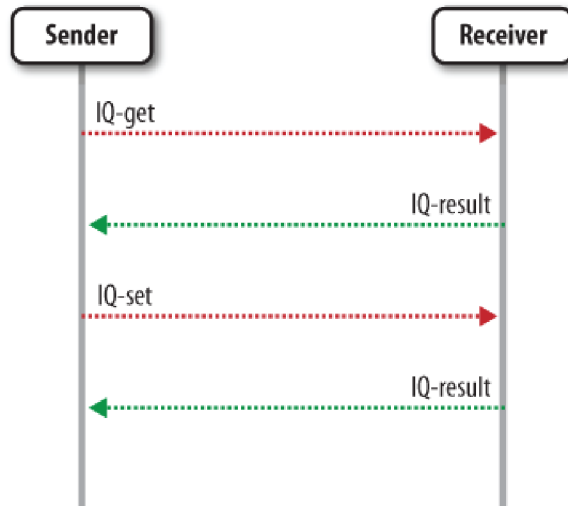
Tento element se chová jako kontejner jak pro odchozí dotazy klienta na server, tak pro příchozí odpovědi ze serveru pro klienta. Prvek může, ale nemusí, obsahovat tyto atributy:

- `to` — Specifikuje adresáta (JID) dotazové/odpovědní stanzy. Tento atribut je pro většinu dotazů volitelný, např. když je adresát server. Může se však stát, že dotaz není směrován přímo serveru, ale nějaké jeho komponentě, a tudíž je potřeba adresáta upřesnit.
- `from` — Specifikuje odesílatele (JID) stanzy. Tento atribut je též volitelný, ale je potřeba zadávat, pokud komunikujeme s komponentou serveru nebo někým jiným, než je server, abychom nemohli předstírat falešnou identitu (pokud je zadáno jiné JID, než naše, server to přepíše).
- `id` — Volitelná vlastnost využitelná pro záměry sledování určitých posloupností dotaz-odpověď. Vyplňuje se zde náhodný řetězec znaků.
- `type` — Nutný atribut, který specifikuje účel dotazu/odpovědi. Může být jeden z těchto možností: `get`, `set`, `result` a `error`. Možnosti `get` a `set` jsou, jak název napovídá, na žádost o informaci a nastavení dat, zatímco `result` a `error` jsou odpovědní v případě kladné popř. záporné odpovědi.

Obsah všech dotazových/odpovědních stanz závisí nejvíce na jmenném prostoru. Na obrázku 3.1 je ukázána typická výměna dat při info/query dotazu.

---

<sup>7</sup><http://xmpp.org/resources/schemas/>



Obrázek 3.1: Typická komunikace iq stanzou [6]

### Prvek <presence>

Mimo užití pro klientskou dostupnost a prezenci je tento prvek použit také serverem jako nástroj pro management připojených klientů. Výčet jeho atributů je stejný jako u elementu <iq>. Akorát výčet možností atributu `type` je rozdílný. Obsahuje:

- `available` — Dostupný, implicitní hodnota, není nutno zadávat atribut `type`.
- `unavailable` — Posílá server, když se klient odpojí – není již dostupný pro komunikaci.
- `probe` — Požadavek o zjištění stavu. Zasílá server ostatním klientům, když se připojí klient a požaduje dostupnost uživatelů z *rosteru*<sup>8</sup>, lze zaslat i z klienta na klienta, ale je potřeba mít povoleno sdílení stavu.
- `subscribe` — Požadavek na povolení o odebírání stavu. Stanza musí obsahovat `from` a `to` atributy.
- `unsubscribe` — Požadavek na odepření odebírání stavu. Jinak stejné.
- `subscribed` — Potvrzení o povolení o odebírání stavu. Odpověď na `subscribe`, musí obsahovat `from` a `to` atributy.
- `unsubscribed` — Odpověď na `unsubscribe`, musí obsahovat `from` a `to` atributy.
- `error` — Chybové hlášení kvůli přenosu nebo zpracování stanza.

Kromě hlavních atributů se používají subelementy pro upřesnění vlastností prezence a stavu. Jsou to <show/> (uvádí jednu z možných voleb: `away`, `chat`, `dnd` – Do Not Disturb, `normal` a `xa` – eXtended Away), dále pak subelement <status/> (vlastnoruční popis stavu např. „Vařím oběd.“) a konečně subelement <priority/>, který udává číselnou hodnotu priority (-127 až 128) daného zdroje, podle níž se určuje, kam doručit zprávu v případě vícero zdrojů. Když je číslo priority menší než 0, zprávu dostane pouze za použití celého JID i se zdrojem a nikoliv jen za použití holého JID. Kromě těchto „hlavních“ subelementů existuje ještě subelement <error/>, který obsahuje jeden z chybových kódů protokolu. Tento subelement se používá, když je atribut `type` nastaven na `error`.

<sup>8</sup>Seznam kontaktů jabberu

## Prvek <message>

Tento prvek je hlavním nositelem zprávy a používá zase stejné atributy jako předcházející dva prvky, akorát má povinně atribut `to` a subelement `<body>`. Atribut `from` je doplněn serverem podle toho, kým byla zpráva odeslána. Subelement `<body>` je párovým tagem, který ale nesmí již mít žádné další atributy. Naopak může obsahovat vnořený subelement `<x/>`, jenž zpravidla obsahuje další informace o zprávě jako např. jestli byla zpráva doručena klientovi, jestli byla zobrazena, jestli uživatel stále píše zprávu, kdy byla zpráva odeslána a jiné. Interpretace těchto informací závisí na daném jmenném prostoru.

Protože je XMPP protokol založený na XML, není možné používat všechny znaky v textu zpráv a některé je třeba vložit do tzv. CDATA (character data) sekce, zanořením textu do tagů `<![CDATA[ a ]]>`, nebo je zakódovat escape sekvencemi<sup>9</sup>. Jako příklad escapování mohu uvést znaky `&`, `<` a `>`, které se kódují jako „`&amp;`“, „`&lt;`“ a „`&gt;`“.

Dalšími vnořenými subelementy jsou:

- `<subject/>` — Vyjadřuje předmět zprávy, nepoužívá se pro chat.
- `<thread/>` — Značí vlákno konverzace pro zpětné trasování konverzací.
- `<error/>` — Obsahuje atribut `code` pro kód chyby a může obsahovat popis chyby v přirozeném jazyce.

Čerpáno z [5].

### 3.8.5 Zahájení sezení a autentizační mechanismy

Každé sezení je vlastně jeden předem autentizovaný a „spoutaný“ (bind — viz dále) stream, zašifrovaný, či nikoliv. K šifrování se používá TLS a k autentizaci některý mechanismus SASL<sup>10</sup>.

Hlavní SASL mechanismy:

- PLAIN — Nejjednodušší SASL mechanismus, obsahuje login a heslo oddělené 0 bytem v kódování *Base64*, doporučuje se používat jen v TLS módu.
- DIGEST-MD5 — Složitější šifrování, šifruje se za pomoci challenge, není potřeba TLS spojení, ale vyvstaly problémy s tímto šifrováním a proto ho IETF odmítá ve prospěch mechanismu SCRAM.
- SCRAM — Salted Challenge Response Authentication Mechanism, náhrada za DIGEST-MD5, poskytuje silnou ochranu ve spojení s TLS, upravuje ho RFC 5802<sup>11</sup>.
- EXTERNAL — Tento mechanismus umožňuje použití certifikátu ve TLS vyjednávání místo hesla, ale velmi málo koncových uživatelů mimo specializovaná prostředí vlastní osobní certifikát, a tak se tento způsob autentizace používá více k ověření serveru serverem.
- GSSAPI — Používá se v systémech KERBEROS V např. v korporátních společnostech.
- ANONYMOUS — Umožňuje autentizaci bez registrace, účet pro jedno použití (call centrum, online pomoc,...).

<sup>9</sup><http://cs.wikipedia.org/wiki/Escapov%C3%A1n%C3%AD>

<sup>10</sup>Simple Authentication and Security Layer

<sup>11</sup><http://tools.ietf.org/html/rfc5802>

Postup vytvoření nového sezení klient-server (na příkladu Google Talku) je následující:

1. Klient zašle stream.

```
<stream:stream to="gmail.com"
xmlns:stream='http://etherx.jabber.org/streams'
xmlns='jabber:client' version='1.0'>
```

2. Server odpoví stream a možnosti šifrování a autentizace.

```
<stream:stream from="gmail.com" id="5713462266C17E88"
version="1.0" xmlns:stream="http://etherx.jabber.org/streams"
xmlns="jabber:client">
  <stream:features>
    <starttls xmlns="urn:ietf:params:xml:ns:xmpp-tls">
      <required/>
    </starttls>
    <mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl">
      <mechanism>X-GOOGLE-TOKEN</mechanism>
      <mechanism>X-OAUTH2</mechanism>
    </mechanisms></stream:features>
```

Příčemž subelement `<required/>` znamená, že šifrování přes TLS je nutné.

3. Klient zašle požadavek na spojení TLS.

```
<starttls xmlns='urn:ietf:params:xml:ns:xmpp-tls' />
```

4. Server odpoví.

```
<proceed xmlns="urn:ietf:params:xml:ns:xmpp-tls"/>
```

5. Ustaví se připojení pomocí TLS a znovu klient zašle stream.

```
<stream:stream to="gmail.com"
xmlns:stream='http://etherx.jabber.org/streams'
xmlns='jabber:client' version='1.0'>
```

6. Server odpoví zase s možnostmi autentizace v šifrovaném módu.

```
<stream:stream from="gmail.com" id="24ED827D33B378F8" version="1.0"
xmlns:stream="http://etherx.jabber.org/streams" xmlns="jabber:client">
  <stream:features>
    <mechanisms xmlns="urn:ietf:params:xml:ns:xmpp-sasl">
      <mechanism>PLAIN</mechanism>
      <mechanism>X-GOOGLE-TOKEN</mechanism>
      <mechanism>X-OAUTH2</mechanism>
    </mechanisms></stream:features>
```

7. Klient posílá vybraný mechanismus SASL s danými přihlašovacími údaji.

```
<auth xmlns='urn:ietf:params:xml:ns:xmpp-sasl' mechanism='PLAIN'>
AGhham55agBoam5qNzg5NjU0MTIzjj</auth>
```

Jestliže je použit jiný mechanismus, může klient ještě čekat na tzv. *challenge* od serveru a komunikace může vypadat jinak.

8. Server odpovídá, jestli se přihlášení zdařilo.

```
<success xmlns="urn:ietf:params:xml:ns:xmpp-sasl"/>
```

9. Klient zašle znovu stream.

```
<stream:stream to="gmail.com"
xmlns:stream='http://etherx.jabber.org/streams'
xmlns='jabber:client' version='1.0'>
```

10. Server odpoví požadavkem na spoutání klientu k určitému zdroji.

```
<stream:stream from="gmail.com" id="D791C768590CD836" version="1.0"
xmlns:stream="http://etherx.jabber.org/streams" xmlns="jabber:client">
  <stream:features>
    <bind xmlns="urn:ietf:params:xml:ns:xmpp-bind"/>
    <session xmlns="urn:ietf:params:xml:ns:xmpp-session"/>
  </stream:features>
```

11. Klient odpoví, k jakému zdroji chce být připoután.

```
<iq type="set" id="1">
  <bind xmlns="urn:ietf:params:xml:ns:xmpp-bind">
    <resource/>
  </bind>
</iq>
```

Zde je uvedeno, že o jménu zdroje má rozhodnout server, jinak by byl tag `<resource>` párovým a mezi něj by byl vložen požadovaný název zdroje

12. Server odpoví se jménem zdroje ve formátu plného JID.

```
<iq id="1" type="result">
  <bind xmlns="urn:ietf:params:xml:ns:xmpp-bind">
    <jid>hajnyj@gmail.com/C402623C</jid>
  </bind>
</iq>
```

13. Klient zašle oznámení o započatí sezení.

```
<iq type='set' id='2'>
  <session xmlns='urn:ietf:params:xml:ns:xmpp-session' />
</iq>
```

Příčemž toto je zastaralé a server nabízí session jen z důvodu zpětné kompatibility. Když klient nezašle oznámení o započatí sezení, server pokračuje jakoby se tak stalo[6].

14. server odpoví

```
<iq type="result" id="2"/>
```

15. Klient započne sezení žádostí o roster a zasláním presence stanzy.

```
<iq from="hajnyj@gmail.com/C402623C" type="get" id="3">
  <query xmlns="jabber:iq:roster"/>
</iq>

<presence/>
```

### 3.8.6 Rozšíření

Původní vývojáři Jabberu byli zaměřeni pouze na výstavbu IM systému. Avšak možnosti rozšiřitelnosti XML ztraktivnili XMPP pro vývojáře, kteří potřebují spolehlivou infrastrukturu pro rychlou výměnu strukturovaných dat a ne jen IM služby. Proto se XMPP používá na vývoj široké škály aplikací zahrnující syndikaci obsahu (znamená opětovné dodávání aktuálních informací, které obsahuje internet), varování a notifikace, odlehčené middleware (software, který slouží jako konverzní nebo překladatelská vrstva) a webové služby, sdílení tabule, vyjednávání o multimediálním sezení, geolokace, online hraní, sociální sítě a další.

Roky byla navrhována různá rozšíření pro XMPP, tzv. *XEP* (XMPP Extension Protocol)<sup>12</sup>. Tato rozšíření tvoří spolu s jádrem protokolu základní stavební kámen pro rozšíření podle představ každého jednotlivce. Pokud někomu něco v XMPP specifikaci a rozšíření chybí, může jednoduše vytvořit svoje rozšíření a podstoupit otevřený proces standardizace.

### 3.8.7 Výhody a nevýhody

Hlavními přednostmi XMPP oproti jiným protokolům jsou:

- *otevřenost* — Na protokol se nevztahuje žádná komerční licence a je ho tedy možné používat k libovolným účelům a dokonce ho dle vlastních potřeb modifikovat.
- *standardizovanost* — Organizace IETF prohlásila protokol za standard a ten byl popsán v RFC 3920 a RFC 3921.
- *bezpečnost* — Protokol umožňuje uzavřít síť vůči jiným sítím, díky čemuž mohou uživatelé komunikovat např. pouze v rámci jedné domény. Navíc do samotného jádra protokolu je včleněno šifrování pomocí technologie SASL a kryptografického protokolu TLS.
- *decentralizovanost* — Podobně jako například u e-mailu neexistuje žádný jediný řídicí server, ke kterému by se museli všichni uživatelé přihlašovat. Každý si může na svůj počítač nainstalovat XMPP server a dát ho k dispozici uživatelům. V kombinaci s možností uzavřít síť vůči ostatním sítím je tedy XMPP takřka ideálním kandidátem například pro interní komunikační řešení v rámci firem.
- *rozšiřitelnost* — Pokud chce někdo využít protokol v rámci svého projektu, ale nedostačuje mu existující funkcionality, lze komunikaci obohatit o zprávy vlastního typu. Je čistě na klientské aplikaci, jak zprávy zpracuje. Server ani protokol je nijak neomezují.

Mezi další výhody patří například to, že uživatel může být přihlášen na více místech najednou (viz zdroje). Další vlastností je podpora komunikace více uživatelů najednou (tzv. MUC – Multiuser conference), která vypadá obdobně jako komunikace prostřednictvím protokolu IRC. Uživatelé mají k dispozici místnost, ve které probíhá jejich hromadná konverzace. Tato místnost může být veřejná (tedy přístupná pro koholiv) nebo chráněná heslem, které je vyžadováno pro vstup do místnosti. Ve výchozím stavu se zprávy odesílají všem uživatelům. Pokud chtějí dva uživatelé komunikovat jen spolu, je mezi nimi vytvořena zvláštní místnost.

Za nevýhodu lze považovat malé zastoupení mezi uživateli. Velké procento z nich stále ještě používá zažité systémy jako ICQ, AIM, Live Messenger a Yahoo! Messenger.

---

<sup>12</sup><http://xmpp.org/xmpp-protocols/xmpp-extensions/>

## Kapitola 4

# Návrh aplikace IM Client pro BlackBerry PlayBook

### 4.1 Teoretické poznatky pro návrh uživatelského rozhraní na platformě BlackBerry Tablet OS

#### 4.1.1 Omezení při návrhu pro mobilní zařízení

Mobilní zařízení mají následující vlastnosti rozdílé od PC/notebooků:

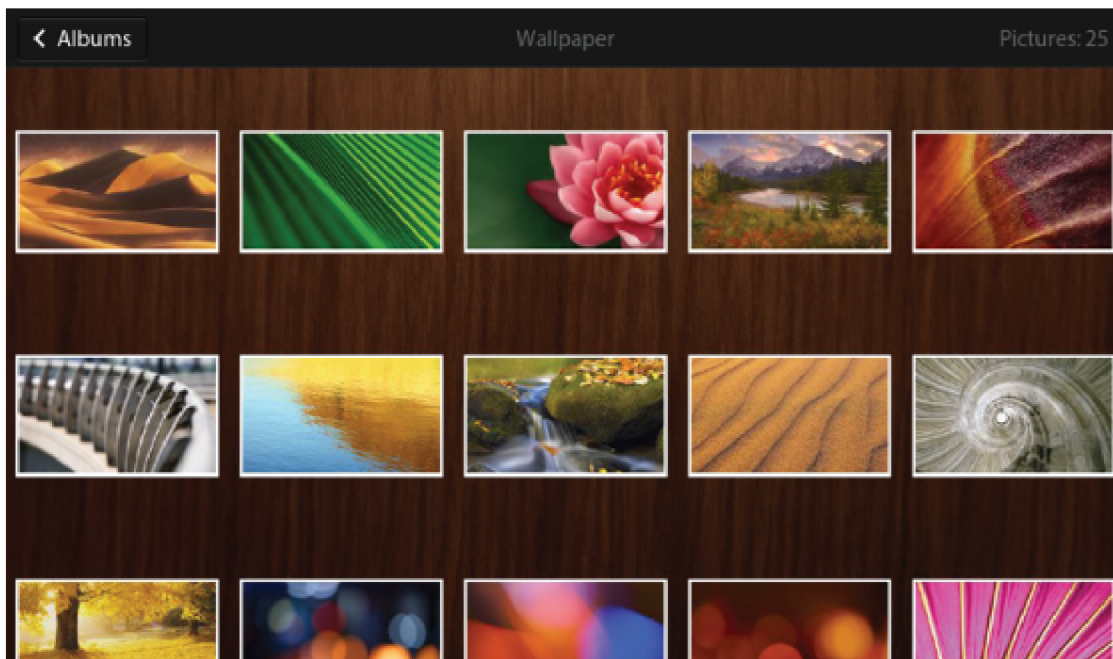
- *Menší velikost obrazovky* — Je třeba zaměřit se pouze na jednotlivé uživatelské úkony a zobrazovat jen informace potřebné uživatelem v daný čas.
- *Nelze zobrazit více obrazovek najednou* — Používat jen jednu obrazovku, kde je to možné. Pokud aplikace potřebuje vícero oken najednou, je třeba rozdělit obrazovku nebo předělat průběh aplikace.
- *Kratší životnost baterií* — Řídit datový tok efektivně, čím méně zařízení přenáší dat, tím větší je jeho výdrž baterie.
- *Bezdrátové připojení do sítě* — Zjednodušit vytváření síťového připojení, ve srovnání se standardními sítěmi LAN jsou zde delší prodlevy, které ovlivňují jak rychle dostanou uživatelé potřebné informace.
- *Pomalejší takt procesoru* — Vyvarovat se procesorově náročných úkonů, aby se aplikace nestala neodpovídající na vstupy.
- *Méně dostupné paměti* — Uvolnit paměť, kdy je to možné. Např. když není aplikace používána, snažit se, aby nepoužívala paměť. O toto se však při použití Adobe AIR stará virtuální stroj a *garbage collector* (automatická správa paměti).

#### 4.1.2 Principy návrhu

##### Návrh pro bezokrajovou obrazovku

Na rozdíl od návrhu pro chytré telefony, kde se vkládá systém obrazovek do hierarchie, by měl mít návrh pro BlackBerry PlayBook plochou strukturu. Místo kontejneru, uvnitř něhož leží celá aplikace, by měla být obrazovka jako okno do aplikace. Např. pokud při pohledu na výčet počet fotek přesahuje rámeček aplikace, měla by se zbývající fota ukázat alepoň z části na okraji obrazovky, aby bylo zřejmé, že uživatelé mohou shlédnout více fotek tažením obrazovky dolů nebo nahoru (tzv. *swipe*). Viz obrázek 4.1.





Obrázek 4.1: Ukázka okna do aplikace. Převzato z [4].

**Použití strategie plátna** Velký prostor rozhraní typické PC aplikace umožňuje uvést mix obsahu a UI komponent. Stejná aplikace pro BlackBerry PlayBook vyžaduje odlišný přístup. Ve většině situací by měla využít aplikace celé obrazovky, používat gesta k orientaci a přistupovat k dalším možnostem jen, když je třeba. Pokud je k prezentování velké množství informací, je nutné použít strategii plátna (*canvas*), což znamená, že obrazovka bude jen rámeček, který obsahuje malý výřez z velkého plátna. Uživatelé pak mohou pohybovat s pohledem tažením prstu v každém směru. Tohoto se dá využít např. v navigaci mapy nebo většího obrázku. Plátno může být buď spojitě, nebo diskrétní. Spojité plátno může být svévolně rozčleněno (např. mapa, plánec stavby). Kdežto diskrétní plátno obsahuje zřejmé, definované subkomponenty (např. seznam kontaktů, eBook). Na spojitěm plátně by mělo být umožněno pohybovat se pomalu, rychle, přibližovat a oddalovat a třeba rotovat. Naproti tomu na diskrétním plátně by měl být umožněn nejen pohyb pomalu a rychle, ale i specifické úkony pomocí gest jako např. pohyb skrz seznam kontaktů nebo skok na další kapitolu v dokumentu eBooku.

### Zaměření na obsah

Je potřeba dělat aplikace jednoduché na použití intuitivními akcemi, když uživatel pracuje s obsahem. Místo použití UI komponent jako např. tlačítka a posuvné lišty, je dobré interagovat přímo s obsahem. Například když uživatel klikne na album, aplikace by měla zobrazit seznam písní v albu. Místo aby se otevřelo kliknutím na album a poté na tlačítko „Otevřít“.

### Zjednodušení funkcionality

Je potřeba znát účel aplikace. Pokud je znám cíl aplikace, je třeba udělat rozhodnutí, co vše patří do základní funkcionality programu. Vše nadstavbové je nunto přesunout do

výsuvného menu nebo úplně odebrat. Pokud se portuje aplikace původně navržená pro PC, zjednodušení není snadné. Měly by se poskytnout pouze ty vlastnosti, které lidi nejvíce zajímají.

### 4.1.3 Návrh obrazovky

BlackBerry PlayBook má na šířku orientovaný displej, a tudíž by se měly aplikace vyvíjet převážně v tomto pohledu. Při navrhování aplikace je nutné myslet na vytvoření širokoúhlého zážitku pro uživatele. Například umístit komponenty UI na strany místo nahoru či dolů v obrazovce, aby se využila výhoda šířky navíc. Je třeba rozlišit primární orientaci aplikace. Je možné vytvořit aplikaci v pohledu na šířku, na výšku, nebo i obojí dohromady, protože BlackBerry PlayBook je navržen, aby souvisle přetočil pohled, pokud se nahne daným směrem (*autoorient*).

Nejlepší postupy:

- Umístit důležité informace na horní okraj obrazovky.
- Vyvarovat se umístění obsahu, kde tomu brání běžné používání.
- Umístit komponenty UI na místa, kde by palce uživatele držely tablet.
- Použít BlackBerry komponenty UI, kde je to možné, aby aplikace zdělila implicitní chování komponent. Uživatelé se pak nemusí učit nové chování komponent implementovaných v aplikaci.
- Vyvarovat se použití uvítací obrazovky (*splash screen*), pokud aplikace není vytížená zdroji a nenačítá se dlouho.

### Nabídky

Úloha nabídky (*menu* — viz obrázek 4.2) je odebrat komponenty UI, jako tlačítka a pole se vstupem, pryč z hlavní obrazovky aplikace. Nabídka může obsahovat funkce, které chceme skrýt, abychom udrželi jednoduchou hlavní obrazovku, nebo nastavení a možnosti, ke kterým se nepřístupuje tak často. Pokud je implementována nabídka, uživatelé by měli mít možnost zobrazit ji gestem „táhnutím z horní části rámečku“.

### Text

Text se používá ke značení předmětů, poskytnutí instrukcí a sdělení jednoduchých představ. Text je jedním z nejdůležitějších aspektů efektivního UI. Návrháři, spisovatelé a vývojáři by měli spolupracovat, aby stvořili efektivní text aplikace. Pokud text není dostatečně dobrý, problém pravděpodobně tkví v návrhu.

Nejlepší postupy:

- Všechn text v UI by měl nést uživatelsky přívětivý jazyk a konverzační tón.
- Vyvarovat se rozvlácnosti. Příliš mnoho vysvětlování odrazuje od čtení a je nekonzistentní s UI tabletu BlackBerry PlayBook. Pokud je třeba vysvětlit komplexní problém, přidává se text do pomocné obrazovky (*Help*).
- Vyvarovat se použití symbolů jako je obchodní značka (<sup>TM</sup>) v UI, místo toho je přidat do obrazovky „O aplikaci“.
- Umísťovat návěští (*Label*) vlevo od komponenty UI, jedinou výjimku zde tvoří zaškrtnutá políčka a skupinová tlačítka, které by měly mít popisky vpravo.



Obrázek 4.2: Ukázka nabídky, převzato z [4].

**Písma** Myriad je implicitní a preferované písmo pro tablet BlackBerry PlayBook, protože je navrženo pro jednoduché čtení pro většinu uživatelů. Je třeba užívat koordinované druhy písma a být konzistentní.

Nejlepší postupy:

- Používat velikost písma 21 pixelů pro normální text a 36 pixelů pro nadpisy, vyvarovat se velikostí písem menší než 15 pixelů.
- Používat standardní šířku písma, středně zúžené písmo jen pro místa s omezenou velikostí.
- Používat kurzívu pro zdůraznění.
- Vyvarovat se podtrženého textu, jedině když vytvoříme hypertextový odkaz s dlouhým řetězcem.

**Kapitalizace a interpunkce** Konzistence je nejdůležitějším aspektem kapitalizace a interpunkce textu. Je nutné věnovat pozornost konkrétním pojmům, frázím a zkratkám, a ujistit se, že se budou zobrazovat stejným způsobem, jako byly napsány.

Nejlepší postupy:

- Nepoužívat dvojtečky k zakončení popisků.
- Vyvarovat se nepotřebnému zakončení tečkou, jen u celých vět. Krátké fráze a seznamy neukončovat tečkou.
- Vyvarovat se používání pouze velkých písmen. Uživatel může získat dojem, že na něj někdo křičí.
- Použít tučné písmo při odkazu na uživatelsky definované jméno.
- Vyvarovat se použití výpustky (...), kromě naznačení zkráceného textu.
- Použít kapitalizaci ve všech popisících komponent UI kromě zaškrťovacích políček a skupinových tlačítek.

## Dialogová okna

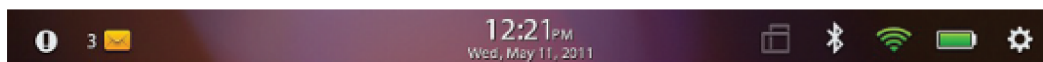
Dialogová okna by se měla používat obezřetně, protože narušují průběh práce uživatele. Místo toho by se měla zobrazit informace společně s předmětem. Například když uživatel nevyplní požadované pole ve formuláři, aplikace by měla identifikovat toto pole a nezobrazovat dialogové okno s chybou. Ten samý přístup se může používat s informací o postupu. Např. když uživatel stahuje aplikaci, je možné zobrazit informaci o postupu v aktuálním kontextu. Tento přístup umožňuje uživateli dokončit jinou úlohu v průběhu stahování.

Nejlepší postupy:

- Používat dialogová okna střídavě. Omezit se na tyto situace:
  - Informovat uživatele o urgentních informacích nebo stavu důležitých akcí.
  - Varovat uživatele o neočekávaných podmínkách nebo situacích, kdy je možná ztráta dat.
  - Dát uživateli k výběru více možností.
  - V případě potřeby uživatelského vstupu pro další krok.
- Pokud je to možné, použít model implicitního uložení. Například když se nastavují možnosti, aplikovat hned nastavení po změně. Existují výjimky, např. když uživatel zavírá dokument, může se využít dialogové okno na dotaz o uložení změn.

## Upozornění a indikátory

Ikony aplikačních upozornění se objevují v levém horním rohu domácí obrazovky (viz obrázek 4.3). Tato upozornění dávají vědět o aplikačně specifických událostech jako např. příchozí email. Indikátory stavu se zobrazují v pravém horním rohu a ty ukazují stavové informace pro všechny aplikace jako např. stav baterie a bezdrátová připojení. Za běhu aplikace může uživatel vždy tento panel zobrazit gestem tažením z levého nebo pravého horního rámečku. Detaily se zobrazí po kliknutí na jednotlivé ikony.



Obrázek 4.3: Ukázka upozornění a indikátorů, převzato z [4].

### 4.1.4 Navrhování ikony aplikace

Navrhování ikony aplikace nám dává možnost předvést aplikaci uživatelům tabletu BlackBerry PlayBook. V rozhraní marketu BlackBerry AppWorld je ikona první představení aplikace uživatelům. Ikona se též objeví v domácí obrazovce jako způsob otevření aplikace. Je důležité, aby ikona byla vizuálně zajímavá a zároveň aby byla konzistentní se vzhledem ostatních ikon BlackBerry aplikací. Měla by se hodit do seznamu aplikací a nebyť moc mdlá nebo moc výrazná.

Nejlepší postupy:

- Ujistit se, že má ikona následující vlastnosti:
  - Velikost: 86 x 86 pixelů. Pokud bude ikona menší, nezapadne mezi ostatní a pokud bude větší než 90 x 90 pixelů, ani se nezobrazí.
  - Rozlišení: 72 pixelů na palec (PPI - pixel per inch).

- Mód: RGB barvy.
- Typ: PNG.
- Zachovat hlavní grafické prvky v mezích 77 x 77 pixelů. Pokud ikona zabírá celých 86 x 86 pixelů, vypadá moc velká.
- Umístit obrázek ikony 2 pixely od spodní meze plochy 86 x 86 pixelů kvůli správnému zobrazení stínu, jenž je generován automaticky.
- Použít zakulacené obdélníky s radiusem 20 pixelů.
- Aplikovat osvětlení shora dolů a nejvíce uprostřed ikony, aby vypadala korektně se stínem.

#### 4.1.5 Shrnutí

V mojí aplikaci jsem se snažil dbát na tyto upřednostňované postupy. Některé z upřednostňovaných postupů nebylo možné využít. Například upozornění jsem ve své aplikaci nemohl použít z toho důvodu, že zatím v nynější verzi SDK neexistuje takové API, aby se s ním daly upozornění ovládat. V této podkapitole jsem se čerpal z [4].

## 4.2 Zvyklosti uživatelského rozhraní pro instant messaging

Po důkladném prozkoumání uživatelských rozhraní většiny IM služeb (ICQ, AIM, Yahoo! Messenger, Windows Live Messenger) na různých platformách s různě velkými displeji (viz tabulka 4.1) jsem došel k těmto závěrům:

1. K používání typické IM služby je zapotřebí minimálně 2 oken (seznam kontaktů se zobrazením vlastního stavu a okno s konverzacemi), v dalším okně může být např. souhrn zpráv (feedů) z internetu (ICQ, Yahoo! Messenger), hry, nebo po vyvolání se zobrazí do dalšího okna nastavení aplikace.
2. S ohledem na současnou rozšířenost používání širokoúhlých monitorů/displejů je možno tyto dvě okna sloučit do jednoho popř. nechat okno se seznamem kontaktů vždy zobrazené v levé nebo pravé části zobrazené pracovní plochy, protože jeho formát je úzký a vertikálně podlouhlý a tudíž neruší tolik v práci.
3. V mobilních zařízeních menší velikosti (chytré telefony) se používá pro každé jedno okno jedna „obrazovka“ (u androidu se jmenuje aktivita) a horní nebo spodní panel pro přepínání mezi nimi popřípadě vyvolání z nabídky aplikace.
4. V mobilních zařízeních větší velikosti (tablety, netbooky apod.) je možno použít — právě kvůli tomu, že většinou mají širokoúhlé displeje — spojení seznamu kontaktu a seznamu konverzací do jednoho okna. V případě tabletu BlackBerry PlayBook je tomu tak vždy, protože architektura platformy nedovoluje více oken jedné aplikace ani změnu velikosti okna z celoobrazovkového režimu, pouze minimalizaci v podobě miniaturny a v této podobě nelze s aplikací interagovat.
5. Většina IM služeb je typicky zabarvena do nějaké barvy, což dává aplikaci specifický ráz. Například oficiální ICQ klient je zabarven do zelena, oficiální Yahoo! messenger je zabarven do fialova, Windows Live Messenger má implicitní barvy Windows atp. Naproti tomu například oficiální AIM klient používá kombinace barev černých, zašedlých a světlých, což dle mého názoru nemá správný směr, jakým by se mělo uživatelské prostředí ubírat. Volba barev by ale měla být výběrem preferencí každého uživatele, a proto by vhodným krokem mělo být nastavení vzhledu podle uživatele. Názory na tuto problematiku se ale různí.

| Platforma            | Zařízení                | Velikost displeje |
|----------------------|-------------------------|-------------------|
| MS Windows 7         | Notebook + ext. monitor | 22"               |
| Android OS           | Google Nexus S          | 4"                |
| BlackBerry Tablet OS | BlackBerry PlayBook     | 7" širokoúhlý     |

Tabulka 4.1: Tabulka platforem a velikostí displejů u testování služeb IM

## 4.3 Návrh uživatelského rozhraní

Tato podkapitola se zabývá návrhem aplikace pro instant messaging na platformě BlackBerry Tablet OS. Po dohodě s vedoucím práce jsem se zaměřil na protokol XMPP. Důvody jsou popsány výše v kapitole 3. Výsledná aplikace by měla být jednoduchá, stabilní, intuitivně ovladatelná, použitelná a uživatelsky přívětivá. Také právě proto je kladen velký důraz na návrh uživatelského rozhraní. A to ať už na rozvržení komponent, ozvučení, nebo na volbu barevného kontextu. Co se týká vzhledu, klient nepřináší žádné novinky a zapadá do nastoleného trendu mezi klienty pro IM, jelikož nezvyklý vzhled nebo ovládání programu by mohlo být pro uživatele nejasné. Kvůli limitům zmíněných v podkapitole 4.2 musí být výsledná aplikace rozvržena do jednoho okna s přidáním případných nemožných dialogových oken a výsuvné nabídky, která se zobrazí po gestu tažení z horní části dotykového rámečku. Po důkladné analýze všech možných rozvržení jsem zvolil tento případ (viz obrázek 4.4):



Obrázek 4.4: Návrh rozvržení komponent v aplikaci IM Client

### 4.3.1 Ozvučení

Aplikace by měla být řádně ozvučena, aby uživatel měl zpětnou vazbu, když například otevře nový chat, přihlásí se k sezení nebo mu přijde nová zpráva. BlackBerry PlayBook nepodporje vibrační upozornění, protože nemá integrovaný vibrační modul (viz 2.3). Zvuky by měly jít samozřejmě vypnout, pokud to uživatele ruší. Tato vlastnost by se měla dát nastavit v nastavení aplikace ve výsuvné nabídce.

### 4.3.2 Barevný kontext

Barevné rozvržení je jeden z nejdůležitějších aspektů uživatelského rozhraní. Jak je již uvedeno výše, každá aplikace by měla být specifická právě svým barevným kontextem. Proto jsem se rozhodl ve své aplikaci vyladit uživatelské rozhraní netradičně do tmavého stylu. Dominantní barva je černá a doplňuje ji k ní kontrastní oranžová, a obecně světlé barvy. Barevné rozvržení výpisu konverzace se dá upravit jak vybráním jedním ze tří přednastavených schémat (viz obrázky 4.5, 4.6 a 4.7), tak pomocí vlastního „.css“ souboru. Zobrazení vzorového souboru bude zabudováno do aplikace. Po konzultaci s oponentem práce, kde mi bylo doporučeno vybírat barevná schémata z nějaké osvědčené databáze barevných schémat, jako např. [kuler.adobe.com](http://kuler.adobe.com), jsem vybral pro svoji aplikaci tři následující barevná schémata:

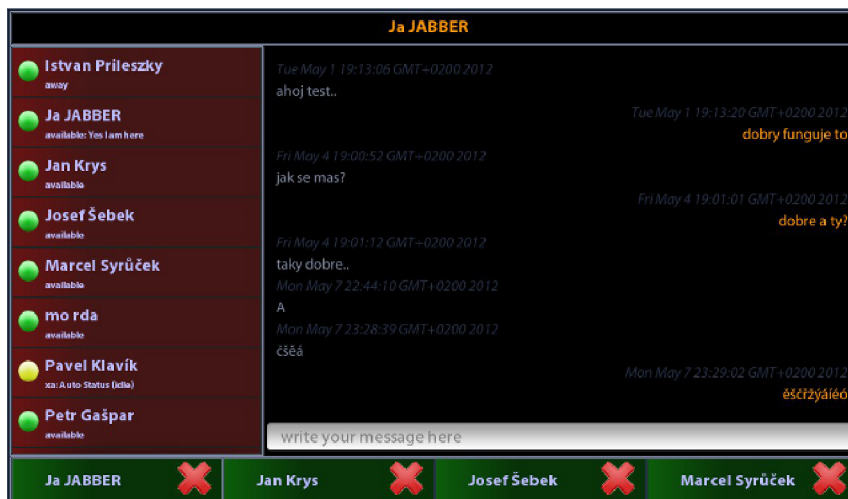


Obrázek 4.5: Barevné schéma výpisu konverzace č.1



Obrázek 4.6: Barevné schéma výpisu konverzace č.2





Obrázek 4.7: Barevné schéma výpisu konverzace č.3

### 4.3.3 Funkce programu

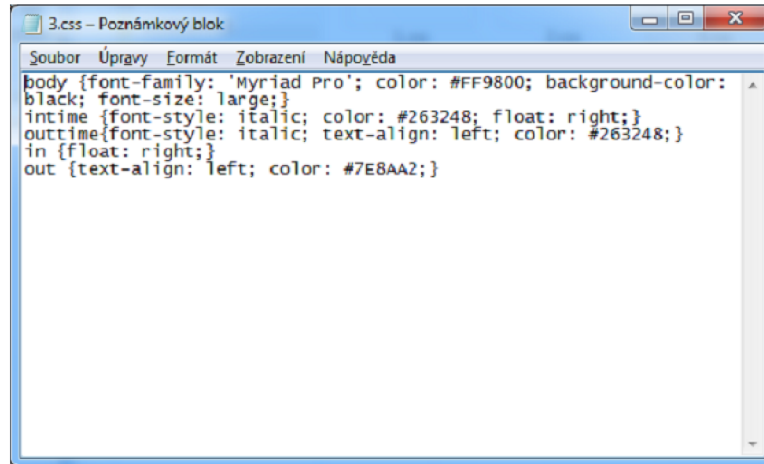
Klient bude splňovat základní funkčnost, která se od klienta pro IM očekává, tedy bude se umět připojit k serveru šifrovaně i nešifrovaně, autentizovat pomocí některých z autentizačních mechanismů SASL, načíst seznam kontaktů ze serveru, přijímat a odesílat zprávy, zobrazovat status uživatelů v seznamu kontaktů a mít souběžně otevřeno několik chatů zároveň. V levé části okna bude zobrazen seznam kontaktů seřazený podle jména s možností nastavení, zda se budou zobrazovat i *offline* kontakty, tzn. i ty, které nejsou přítomny. Každý kontakt bude ikonou a popisem stavu informovat uživatele o stavu kontaktu. Při kliknutí na kontakt se otevře nový chat s již proběhlou historií komunikace. Přepínání mezi jednotlivými chaty by mělo být intuitivní. Každý chat jde křížkem kdykoliv jednoduše vypnout. Při přijetí nové zprávy od kontaktu, se kterým není otevřen chat, se nový chat automaticky otevře.

Nastavení bude umožňovat:

- Zapnutí/vypnutí zvuku.
- Zapnutí/vypnutí zobrazování offline uživatelů.
- Zvolení jednoho ze tří barevných schémat pro stylizovatelný výpis konverzace.
- Zvolení vlastního „.css“ souboru pro barevné schéma stylizovatelného výpisu konverzace.
- Zobrazení vzorové struktury „.css“ souboru (viz obrázek 4.8).
- Vymazání souborů s historií.
- Odeslání zpětné vazby uživatelem. Po kliknutí se zobrazí editace přednastaveného emailu (viz obrázek 4.9)

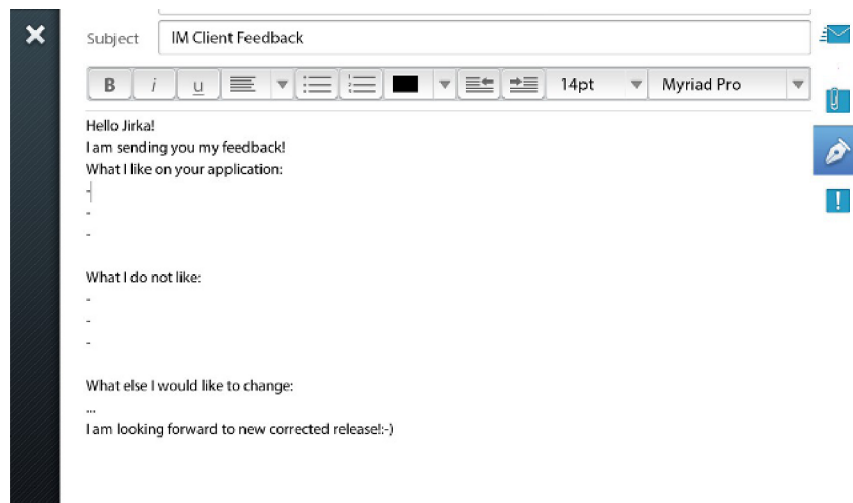
### 4.3.4 Chování programu

Po spuštění aplikace se otevře základní obrazovka bez aktivovaného účtu s nápovědou, jak začít, ve stylizovatelném výpise. Tam je napsáno, aby uživatel nejprve táhnul prstem po obrazovce směrem dolů z horní části dotykového rámečku. Poté se mu objeví výsuvná



```
3.css - Poznámkový blok
Soubor Úpravy Formát Zobrazení Nápověda
body {font-family: 'Myriad Pro'; color: #FF9800; background-color:
black; font-size: large;}
intime {font-style: italic; color: #263248; float: right;}
outtime{font-style: italic; text-align: left; color: #263248;}
in {float: right;}
out {text-align: left; color: #7E8AA2;}
```

Obrázek 4.8: Vzorová struktura CSS souboru pro aplikaci IM Client



Obrázek 4.9: Odeslání zpětné vazby pro aplikaci IM Client

nabídka (viz obrázek 4.10), kde se nachází velké zelené plus, kterým lze přidávat XMPP účty (včetně Facebook chatu a Google Talku). Dále se tam nachází seznam již vytvořených účtů, zaškrtačovací políčko s možností změny zobrazení offline kontaktů a tlačítko pro zobrazení nastavení aplikace. Při použití stejného gesta jako pro zobrazení výsuvné nabídky se nabídka zase schová. Po kliknutí na vytvořený účet se zobrazí nemodální dialogové okno s možnostmi použití účtu. Je možno se připojit, účet editovat, smazat nebo zrušit dialog. Editace účtu je opět intuitivní. Po kliknutí na tlačítko *Připojit* (Connect) se nejprve aplikace odpojí, je-li připojena k nějakému účtu, a poté se připojí k aktuálnímu účtu a stáhne seznam kontaktů ze serveru. V této fázi stačí kliknout na nějaký kontakt pro zahájení konverzace, přičemž se v seznamu chatů objeví nový chat a nahraje se konkrétní soubor s historií chatu ve formátu *HTML*. Pro zrušení chatu se musí kliknout na červený křížek v pravé části okýnka s chatem.



Obrázek 4.10: Základní obrazovka s vysunutou nabídkou v aplikaci IM Client

### 4.3.5 Stylizovatelný výpis konverzace

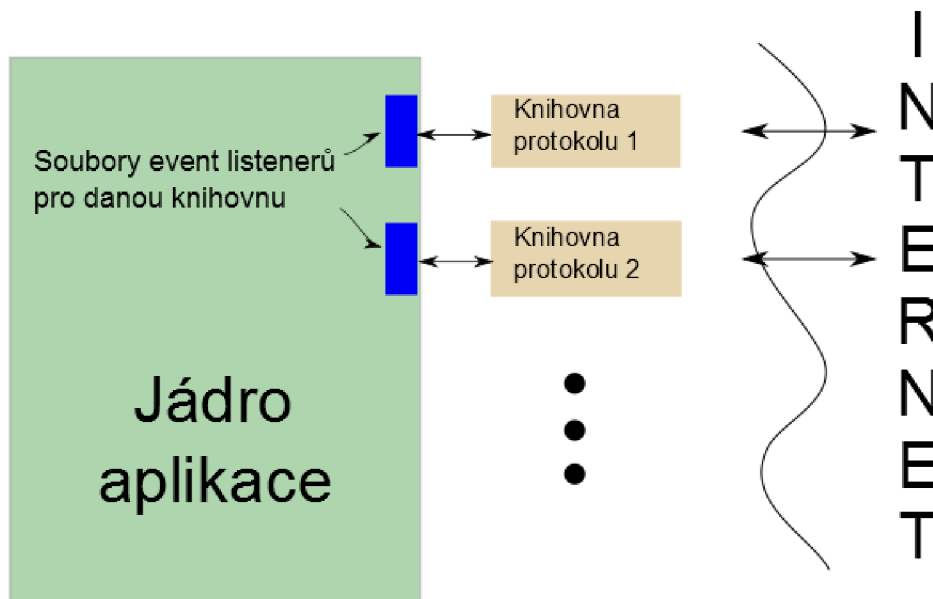
Způsobů, jak vypsát prvky konverzace, existuje mnoho. Já jsem po dohodě s vedoucím práce zvolil výpis ve formě HTML stránky. Tento způsob má mnoho pozitiv i negativ. Mezi hlavní pozitiva patří například snadná skinovatelnost (stylizace) výpisu pomocí souborů CSS, snadné uložení historie a snadná manipulace s obsahem (přibližování a posun textu). Jako hlavní negativum bych uvedl asi nutnost použití komponenty *QNXStageWebView*. Na to jsem ale přišel až během implementace a záporné vlastnosti této komponenty jsou popsány v kapitole 5.

### 4.3.6 Uložené soubory

Program ukládá ve svém *sandboxu* soubory s historií konverzací. Sandbox je abstraktní pojem na pojmenování úložného prostoru aplikace. Soubory jsou pojmenovány podle JID kontaktu a navíc je přidána koncovka „.html“

## 4.4 Návrh struktury aplikace

Struktura aplikace je důležitá pro pozdější rozšíření aplikace o další funkce a hlavně podporované protokoly. Proto by měla být navržena tak, aby se jednoduše a bez velkých změn ve struktuře dalo přidat například rozhraní pro instant messaging pomocí protokolu YMSG, MSNP, nebo třeba i OSCAR. Toho bude docíleno tím, že aplikace je událostmi řízená (*event driven*) a pro každý protokol budou použity jiné události, takže jádro aplikace bude naslouchat a zpracovávat různé druhy událostí. Tento model se dá v prostředí ActionScriptu poměrně jednoduše naimplementovat. Pro ilustraci uvádím ještě obecnou strukturu aplikace (viz obrázek 4.11).



Obrázek 4.11: Návrh obecné struktury aplikace IM Client

## Kapitola 5

# Implementace a testování aplikace

Tato kapitola popisuje konkrétní implementaci aplikace, použití knihoven, jejich úpravu, problémy při implementaci a testování aplikace včetně umístění aplikace na market BlackBerry AppWorld.

Jak již bylo výše uvedeno, implementace probíhala ve vývojovém prostředí Adobe Flash Builder 4.6 v programovacím jazyce ActionScript 3.0. Licence Flash Builderu 4.6 je po zažádání u Adobe <sup>1</sup> pro studenty zdarma pro nekomerční použití. Zároveň byly použity prvky jazyků HTML a CSS (*Cascading Style Sheets*).

### 5.1 Struktura aplikace

Aplikace IM Client se skládá z celkem 14 tříd, rozdělených do 5 balíků (*packages*), včetně Top Level balíku. Balíky jsou rozděleny podle použití obsažených tříd. Jmenují se:

- **containers** — Obsahuje třídy, které dědí ze třídy `qnx.ui.core.Container` a které dále obsahují seznam nějakých subjektů, například seznam účtů. Pro správné explicitní zobrazení seznamu je potřeba implementovat tzv. *item renderer*, což je obecně třída implementující rozhraní `ICellRenderer` nebo přímo zděděná od třídy `CellRenderer`. Seznamy jsou v aplikaci tři, a to seznam kontaktů, seznam účtů a seznam otevřených chatů. K nim bylo potřeba implementovat tyto kontejnery, jejich *item renderery*, objekty hodnot (*Value objects* – viz níže) a události. Dále je zde speciální třída `MenuContainer`, která implementuje všechnu logiku, co se týká výsuvné nabídky a jejích možností.
- **events** — Zde se nacházejí speciální událostní třídy, které dědí ze třídy `flash.events.Event` a obsahují data o kontaktu, účtu a nebo chatu, na který bylo kliknuto. Kliknutím se vytvoří tyto *eventy* a jsou předány posluchačům, kteří je zpracují.
- **itemRenderers** — Třídy, které zajišťují vzhled jednotlivých buněk v seznamu. Překrývají funkci `init()` z rodičovské třídy (inicializace formátu buňky) a implementují metodu `updateCell()`, která je důležitá při vykreslování buněk. Více viz výše.
- **Top Level** — Toto ani není balík jako takový, myslí se tím nejvyšší úroveň v balíkové hierarchii. Obsahuje pouze jednu třídu, a to `IMClient`, která ovládá skoro všechnu aplikační logiku. Tato třída dále obsahuje soubor *event listenerů*, který naslouchá událostem posílaných z knihovny `seismic-as3-xmpp` (o ní viz dále). Tento přístup je navrhován v podkapitole 4.4. Účty se ukládají do tzv. lokálního sdíleného objektu

---

<sup>1</sup><http://www.adobe.com/devnet-apps/flex/free/index.html>

(*local Shared Object*), což je vlastně soubor v aplikačním sandboxu, který uchovává persistentní data ve formátu, který aplikaci vyhovuje.

- **VO** — Zkratka pro *Value objects*, obsahuje pouze výčet veřejných atributů, které se používají pro uložení informací o daných tématech (účet, chat, kontakt). Třída **Contact** je však v této fázi nepoužitelná, protože se místo toho pro zjednodušení používá třída `flash.utils.Dictionary`. Nechal jsem ji pro případ, kdyby se aplikace rozšiřovala o další protokol a bylo by potřeba sjednotit typ *Contact*.

Více implementačních detailů je popsáno v programové dokumentaci na přiloženém DVD.

## 5.2 Použití knihoven

V programu byly použity dvě spolu související knihovny, `seesmic-as3xmpp`<sup>2</sup> a `as3crypto`<sup>3</sup>. Obě však nebyly pro můj program dostatečně vyladěny a proto musely být upraveny. Knihovna `seesmic-as3-xmpp` obsahuje funkce pro práci s protokolem XMPP v jazyce ActionScript 3.0 a používá knihovnu `as3crypto` k připojení přes TLS. Knihovny jsou v projektu přilinkovány ve formě jednoho souboru „.swc“.

### 5.2.1 Úprava knihovny seesmic-as3-xmpp

Tato knihovna umí základní práci s XMPP protokolem, jako je např. ustavení spojení se zadaným serverem – šifrovaně i nešifrovaně, autentizace pomocí jednoduchého autentizačního mechanismu PLAIN a ANONYMOUS, odesílat a přijímat XMPP zprávy, formovat XMPP stanzy z posílaných XML fragmentů, pracovat s JID a obsahuje i základ pro rozšíření *multi-user chat* (víceživatelský chat)<sup>4</sup> a *Publish-Subscribe*<sup>5</sup>. Do těchto rozšíření jsem se nepouštěl z důvodu vysokých požadavků na změnu uživatelského rozhraní. Ale povedlo se mi implementovat rozšíření v podobě autentizačního mechanismu DIGEST-MD5, které je použito při autentizaci Facebook chatu. Byla přidána funkce `handleChallenge()`, která se postará o odeslání zprávné odpovědi na požadavek od serveru, který podporuje DIGEST-MD5.

### 5.2.2 Úprava knihovny as3crypto

Tato knihovna dokáže vykonávat různé šifrovací a kryptologické mechanismy včetně RSA, AES, DES, 3DES, MD5, SHA-1 a mnohé další, a zároveň dokáže vyjednat spojení přes TLS a parsovat certifikáty a formu ASN-1/DER. Parsování ASN-1/DER (které je v mém projektu také potřeba) bylo však v oficiální verzi knihovny chybové, a tak jsem musel najít řešení mezi různými opravenými verzemi na internetu. Našel jsem tři různé úpravy této knihovny a z nich jsem vybíral fungující soubory, dokud vše nefungovalo bez problémů.

## 5.3 Problémy s implementací

Během implementace se vyskytly i jiné problémy než jen s knihovnami, a to především v práci se třídou `qnx.media.QNXStageWebView`, která je dle mého názoru nejhorší komponenta v celém *BlackBerry Tablet OS SDK for Adobe AIR API* (více viz níže). Další problémy přineslo tvoření dialogových oken, které jsem původně zamýšlel mimo jiné pro tvorbu

<sup>2</sup><http://code.google.com/p/seesmic-as3-xmpp/>

<sup>3</sup><http://code.google.com/p/as3crypto/>

<sup>4</sup><http://xmpp.org/extensions/xep-0045.html>

<sup>5</sup><http://xmpp.org/extensions/xep-0060.html>

nových účtů. Ačkoliv existuje základní třída pro tvoření dialogů `qnx.dialog.BaseDialog`, nefungují od ní odvozené třídy. Takže nezbyvá nic jiného, než se spokojit se čtyřmi základními typy dialogů s předem určeným počtem vstupních polí a jejich popisků. Dlouho jsem pro vytváření nových účtů v aplikaci používal dialog `qnx.dialog.PasswordChangeDialog`, který má sice požadovaný počet vstupních polí (4), ale zase nejde změnit zobrazení posledních tří polí jako heslo, tzn. ve formě hvězdiček. To však není vhodné například při zadávání Jabber ID. Po zdlouhavém hledání na fórech jsem zjistil, že to nejde změnit a předělal jsem aplikaci na jiný způsob, než použití dialogových oken.

### 5.3.1 QNXStageWebView

Tato třída slouží k zobrazování HTML obsahu v tzv. *view portu* (výhledovém okně) aplikace. Z mě neznámých důvodů leží vždy v nejvyšší vrstvě aplikace, což s sebou přináší nemalé potíže při potřebě zobrazit jinou komponentu právě nad touto třídou. Například v mé aplikaci bylo potřeba zobrazit výsuvné menu do prostoru, kde se nachází právě `QNXStageWebView`, ale nabídka se vysunovala pod tento prostor a nebylo ji možno vidět celou. Proto jsem musel implementovat změnu *view portu* při každém vysunutí a zasunutí nabídky. To se ukázalo jako jednoduché a funkční řešení.

Dalším problémem bylo doskrolování na konec HTML výpisu při každé příchozí zprávě pro případ dlouhé stránky s obsáhlou historií, aby uživatel nemusel vždy skrolovat až dolů ručně. Z mě neznámých důvodů toto nešlo až do verze OS 2.0.0, kdy tato možnost začala fungovat.

Jako poslední negativum třídy bych uvedl nefunkčnost podtřídy `Scroller`, kterou jsem chtěl použít k záměně nativního přibližování a oddalování HTML stránky (formou jako by to byl obrázek) za zvětšování velikosti fontu v aktuálním css souboru při použití stejného gesta. Tento způsob mi poradil vedoucí práce jako daleko uživatelsky přívětivější. Třída `Scroller` úspěšně odchytává události typu začátek a konec posunu vertikálního i horizontálního a začátek přibližování a oddalování. Po začátku přibližování ale přestane reagovat a konec už neodchytne. Dokud nebude tato komponenta opravena, nemůže být navržený způsob jinak implementován.

## 5.4 Testování

Testování je bezpochyby jednou z nejdůležitějších částí ve vývoji softwarového produktu, a proto je na to třeba klást velký důraz. V mém případě probíhalo debugování přímo na tabletu BlackBerry PlayBook, který vlastním, tudíž jsem nemusel stahovat žádné simulátory a vše potřebné jsem mohl hned ověřovat na reálném zařízení. Další výhodou obecně oproti simulátoru je bezesporu rychlost, přičemž u simulátoru je prostředí virtuální, což znamená, že závisí na taktu procesoru a paměti počítače, na němž simulátor běží. A v neposlední řadě je nevýhodou, že nelze nasimulovat úplně všechny situace jako například vícedotyková gesta anebo údaje z gyroskopu a magnetometru.

Podle internetových stránek společnosti RIM <sup>6</sup> si 90 procent všech majitelů PlayBooku povýší operační systém na tu nejvyšší verzi během měsíce od jejího data vydání. To znamená, že víceméně odpadá nutnost se zabývat v aplikaci s verzemi operačního systému a stačí cílit na nejnovější verzi platformy. Povýšení se vydávají pro všechny PlayBooky globálně najednou, takže stačí aplikaci otestovat na mém aktualizovaném tabletu a je možné

<sup>6</sup><https://bdsc.webapps.blackberry.com/devzone/appworld>

předpokládat, že bude s největší pravděpodobností fungovat i na ostatních tabletech BlackBerry PlayBook.

Protože je v mém projektu zapotřebí ke komunikaci minimálně dvou subjektů, musel jsem pro testování mít zaregistrovány alespoň dva XMPP účty. Zaregistroval jsem si tedy účty `hajnyj@jabber.org` a `hajnyj@jabber.cz` na dvou serverech, které vyplývají ze zaregistrovaných Jabber ID — `jabber.org` a `jabber.cz`. Dále jsem použil k testování můj účet na Google Mail, přes který funguje při správném nastavení i protokol XMPP, a můj účet na Facebook chat.

#### 5.4.1 Umístění aplikace na market BlackBerry AppWorld

Pro přístup do marketu BlackBerry AppWorld není nutné složit žádný vstupní poplatek. Po zaregistrování jako *vendor* (prodejce) se zpřístupní rozhraní pro nahrávání vlastních aplikací, které jsou pak distribuovány mezi uživatele pomocí aplikace AppWorld v systému. Vývojář má možnost upravit jméno aplikace, kategorii, ve které se aplikace zobrazí, dále pak hodnocení, licenci (tzn. jestli bude aplikace placená nebo volně ke stažení), kontaktní email na podporu, krátký a dlouhý popis aplikace, ikonu programu, přidat náhledy aplikace a vybrat, pro které operátory a země bude aplikace dostupná. Náhledy jednotlivých obrazovek se pak uživatelům v aplikaci AppWorld přímo zobrazují. Systém dává vývojáři jednoduchou zpětnou vazbu o užívání dané aplikace. Může zde být zobrazen celkový počet stažení aplikace pomocí tzv. *reportů* a také graf se zobrazením počtu stažení v jednotlivé dny v nastaveném rozmezí dat. Dále je zde k dispozici přehled hodnocení aplikace (*reviews*), které je možno zamítnout, když je aplikace např. velmi podhodnocena. Jednotlivé aplikace je možné stáhnout z distribuce či smazat úplně. Pokud má uživatel nainstalovanou starší verzi a vývojář nahraje novou verzi na BlackBerry AppWorld, zobrazí se uživateli informace, že existuje nová verze pro danou aplikaci a nabídne možnost si ji nainstalovat.

#### 5.4.2 Zpětná vazba

První zveřejnění proběhlo 3.3.2012. Od té doby začaly přibývat recenze, z nichž většina byla negativní, protože aplikace ještě neměla vyladěný vzhled a správnou funkčnost, a proto byly později odmítnuty. Od verze 1.0.5 (schváleno 2.4.2012), kdy byl přidán úvodní dialog pro zaslání zpětné vazby, se začaly pomalu objevovat emaily se zpětnou vazbou. Celkem mi zatím došlo dvanáct zpráv se zpětnou vazbou, z nich však polovina neměla nic vyplněno a na další email již autoři neodpověděli. Ostatní zpětné vazby byly většinou pozitivní. Někteří uživatelé se např. ptali, jak se připojit ke službě Google Talk. Jiný si stěžoval na špatný vzhled klávesnice. Tomu jsem odpověděl, že toto bohužel zatím nejde implementačně změnit. Další si stěžoval na zelenou barvu, kterou aplikace měla až do verze 1.1.2 (schváleno 22.4.2012). Na zelenou barvu si stěžoval i oponent práce a tudíž jsem zvolil nakonec mírně odlišné barevné schéma, které je obsaženo i v nejnovější verzi 1.1.3 (schváleno 7.5.2012). Z těch nejnovějších si jeden uživatel stěžoval na chybějící skupinový chat a rozdělení kontaktů do skupin, což by mohlo být vhodné rozšíření do budoucna. Přitom vychválil rychlost aplikace a uživatelské rozhraní. A poslední uživatel chtěl zprovoznit hlasové a video volání přes Google Talk. Na to však zatím neexistují funkční knihovny a jejich tvorba by zabrala nemalé množství času.

Všechny zpětné vazby i s jejich odpověďmi jsou přiloženy na DVD v podobě souborů „.eml“, které se dají importovat do většiny emailových klientů.



Protože jsem zprvu po vytvoření aplikace neměl žádnou zpětnou vazbu, udělal jsem menší průzkum mezi studenty Fakulty informačních technologií v Brně a získal jsem několik podnětů, se kterými jsem se dále zabýval. Mezi ně patří:

- Náhrada původního dialogu na vytvoření účtu. Kvůli zobrazování hvězdiček místo čitelného textu (viz 5.3).
- Přidání zvuků.
- Odstranění chyby, kdy po vypnutí chatu se stále dalo napsat kontaktu zprávu.
- Přidělení ikon účtů GTalk, Jabber a Facebook chat.
- Vytvoření několika CSS souborů jako vzorová schémata.
- Seřazení kontaktů dle jména. Dříve byly neseřazené.
- Přidání tlačítka cancel do dialogových oken.

Všechny tyto podněty jsem do aplikace zaimplementoval.

Hodně podnětů jsem dostal také po konzultaci s vedoucím práce. Například náhradu zoomování HTML výpisu konverzace za zvětšování a zmenšování písma (viz 5.3). Dále bych uvedl jeho připomínky na celkový vzhled uživatelského rozhraní. Původně aplikace používala více typů písem, což nebylo vhodné. Diference v typech písma byly způsobeny použitím dvou typů tříd pro popisek, a to nativního `qnx.ui.text.Label`, který má implicitní font správně *Myriad Pro*, a obecného `flash.text.TextField`, který má implicitní font *Times New Roman*. Tuto nesrovnalost jsem vyřešil náhradou všech popisků za správný `qnx.ui.text.Label`, což dělá lepší dojem z aplikace, protože je více podobná nativním původním aplikacím.

## Kapitola 6

# Závěr

V této práci jsem se blíže seznámil s vývojovým prostředím Flash Builder 4.6 a vyzkoušel si práci na větším projektu, který se zabývá jak pochopením problematiky jednoho z IM protokolů a následným včleněním jeho knihoven do struktury navržené aplikace, tak i návrhem a úpravami uživatelského rozhraní na základě teoretických poznatků a zpětné vazby uživatelů aplikace. Po dohodě s vedoucím práce byla práce, oproti původnímu zadání (tvorba multi-protokolového klienta), zaměřena na otevřený a standardizovaný protokol XMPP/Jabber, nad kterým fungují i velice rozšířené komunikační sítě jako např. Google Talk nebo Facebook chat. Přínosem tohoto projektu je neoddiskutovatelně právě použitelnost, už jenom počet stažení (ke dni 9.5.2012 je to 2398 viz příloha „report.csv“ a „chart.png“ na příloženém DVD) napovídá docela velké oblibě mezi uživateli. Dalším důkazem oblíbenosti jsou některé pozitivní zpětné vazby viz podkapitola 5.4.

Výsledná aplikace je tedy XMPP klient, který se umí připojit k serveru, navázat s ním šifrované spojení (pokud ho server podporuje), autentizovat se a začít komunikovat s jinými klienty. Za rozšíření považuji podporu autentizačního mechanismu DIGEST-MD5, nad kterým funguje mimo jiné právě Facebook chat, a který nebyl v použité knihovně obsažen.

Zdrojové kódy mé práce jsem uvolnil pod licencí GNU GPL v3 a uveřejnil je na internetové stránce <http://sourceforge.net/projects/imclient/>, odkud je lze volně stahovat.

Do budoucna bych se chtěl ve volném čase dále věnovat práci na tomto klientovi. I když je aplikace docela dobře použitelná, je i nadále co zlepšovat. Menšími změnami by byla například ve zpětné vazbě navrhovaná implementace skupin v rosteru nebo implementace skupinového chatu. Mezi větší změny bych zařadil přidání podpory pro nějaký další protokol, například YMSG nebo OSCAR.

# Literatura

- [1] Hrazdil, J.: Jabber/XMPP robot pro vyhledávání pomocí Google. 2007, bakalářská práce.
- [2] J. Oikarinen, D. R.: RFC1459:Internet Relay Chat Protocol. 1993-05-01 [cit. 2012-04-02].  
Dostupné z WWW: <http://tools.ietf.org/html/rfc1459>
- [3] [překlad Kristýna Konopková], A. C. T.: *ActionScript 3.0:oficiální výukový kurz*. Oficiální výukový kurz, Computer Press, první vydání, 2011, ISBN 978-80-251-3335-4.
- [4] Ltd., R. I. M.: BlackBerry PlayBook Tablet UI Guidelines. 2012-02-15, swd-20120215143913505.  
Dostupné z WWW:  
[http://docs.blackberry.com/en/developers/deliverables/39471/BlackBerry\\_PlayBook\\_Tablet-UI\\_Guidelines-1320689728596-2.0-en.pdf](http://docs.blackberry.com/en/developers/deliverables/39471/BlackBerry_PlayBook_Tablet-UI_Guidelines-1320689728596-2.0-en.pdf)
- [5] Moore, D.; Wright, W.: *Jabber Developer's Handbook*. Developer's Library, Sams, 2004, ISBN 9780672325366.  
Dostupné z WWW: <http://books.google.cz/books?id=9tl6QgAACAAJ>
- [6] SAINT-ANDRÉ, K. S. a. R. T., Peter: *XMPP:the definitive guide : building real-time applications with jabber technologies*. O'Reilly, první vydání, 2009, ISBN 978-0-596-52126-4, 287 s.
- [7] WWW stránky: Windows Live Messenger, a short history. 2010-02-09 [cit. 2012-04-02].  
Dostupné z WWW: [http://windowsteamblog.com/windows\\_live/b/windowslive/archive/2010/02/09/windows-live-messenger-a-short-history.aspx](http://windowsteamblog.com/windows_live/b/windowslive/archive/2010/02/09/windows-live-messenger-a-short-history.aspx)
- [8] WWW stránky: RIM Launches Next Generation BlackBerry Web Application Platform, Releases Code to Open Source Community. 2010-09-27 [cit. 2012-03-28].  
Dostupné z WWW: <http://press.rim.com/release.jsp?id=4576>
- [9] WWW stránky: Microsoft Notification Protocol - Wikipedia. 2011-11-17 [cit. 2012-03-29].  
Dostupné z WWW:  
[http://en.wikipedia.org/wiki/Microsoft\\_Notification\\_Protocol](http://en.wikipedia.org/wiki/Microsoft_Notification_Protocol)
- [10] WWW stránky: Join us at BlackBerry 10 Jam for hands-on time with BlackBerry 10 tools and... 2012-03-23 [cit. 2012-03-26].  
Dostupné z WWW:  
<http://blogs.blackberry.com/2012/03/blackberry-10-jam-tools/>

- [11] WWW stránky: BlackBerry - Playbook Specs. 2012 [cit. 2012-03-27].  
Dostupné z WWW: <http://us.blackberry.com/playbook-tablet/tablet-specs.jsp>
- [12] WWW stránky: Sample apps - BlackBerry Native. 2012 [cit. 2012-03-27].  
Dostupné z WWW: <https://bdsc.webapps.blackberry.com/native/sampleapps>
- [13] WWW stránky: Downloads - BlackBerry HTML5/WebWorks. 2012 [cit. 2012-03-28].  
Dostupné z WWW: <https://bdsc.webapps.blackberry.com/html5/download>

# Příloha A

## Obsah CD

Tato kapitola popisuje obsah CD nosiče:

- **app/** — V této složce se nachází aplikace jako projekt ve Flash Builderu 4.6 (složka `IMClient/`) včetně programové dokumentace v podsložce `doc/`, instalačního souboru `IMClient.bar` a zdrojových kódů v podsložce `src/`. Dále je zde nahrána i knihovna `seismic-as3xmpp` (složka `seismic-as3-xmpp`), která je potřeba pro běh aplikace.
- **doc/** — Zde se nachází vlastní text bakalářské práce: soubor `zhajny02.pdf` a zdrojové soubory pro systém `LATEX`.
- **other/** — Zde se nachází přílohy jako report počtu stažení na BlackBerry AppWorldu (soubor `report.csv` a `chart.png`), počet řádků zdrojových souborů (soubor `totalLOC.txt`) a soubory zpětné vazby ve složce `feedback`.
- **poster/** — Zdrojové soubory doprovodného plakátu ve formátu `svg`, včetně fontů a použitých obrázků, a výsledný plakát ve formátu `png`.

**Příloha B**

**Plakát**