

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SDÍLENÁ TABULE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAROSLAV ŘEZNÍK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SDÍLENÁ TABULE

SHARED BOARD

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAROSLAV ŘEZNÍK

VEDOUcí PRÁCE

SUPERVISOR

Doc. Ing. TOMÁŠ VOJNAR, Ph.D.

BRNO 2008

Abstrakt

Cílem projektu “Sdílená tabule” je navrhnout a implementovat aplikaci, která umožní spolupráci na skupinových projektech v reálném čase v síťovém prostředí. Komunikace může probíhat jak pomocí textových zpráv, tak multimediálně pomocí kreslení na sdílené tabuli, vkládáním interaktivních objektů a hlasovou či video komunikací.

V rámci semestrální části projektu se řeší využití interaktivních tabulí, návrh komunikačního protokolu a grafického uživatelského rozhraní. V diplomové práci následně implementace aplikace.

Klíčová slova

sdílená tabule, interaktivní tabule, protokol XMPP, síťová spolupráce, kreslení

Abstract

The aim of the “Shared Board” project is to design and implement an application that will allow a real-time collaboration on workgroup projects in a network environment. The communication can be accomplished using Instant Messaging or multimedia whiteboard with a possibility of drawing, attaching interactive objects and combined with voice and video communication.

The term project part solves utilization of interactive whiteboards, design of communication protocol and graphical user interface. Implementation is solved as masters thesis part.

Keywords

shared board, interactive board, XMPP protocol, network collaboration, drawing

Citace

Jaroslav Řezník: Sdílená tabule, diplomová práce, Brno, FIT VUT v Brně, 2008

Sdílená tabule

Prohlášení

Prohlašuji, že jsem tuto semestrální práci vypracoval samostatně pod vedením pana Ing. Radka Vokála a pana Doc. Ing. Tomáše Vojnara, PhD.

.....
Jaroslav Řezník
15. května 2008

Poděkování

Děkuji pánům Doc. Ing. Tomáši Vojnarovi a Ing. Radkovi Vokálovi za vedení diplomové práce a společnosti Red Hat Czech za možnost pracovat na tomto zajímavém projektu.

© Jaroslav Řezník, 2008.

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

1 Úvod	5
1.1 Slovo úvodem	5
1.2 Historie počítačové komunikace v reálném čase	5
1.3 Současnost	6
1.4 Cíl projektu	6
1.5 Obsah technické zprávy	6
1.5.1 Semestrální projekt	7
2 Interaktivní tabule	8
2.1 Technické vybavení pro interakci s počítačem	8
2.1.1 Běžná polohovací zařízení	8
2.1.2 Tablety	8
2.1.3 Dotykové obrazovky	8
2.1.4 Specializované systémy interaktivních tabulí	9
2.2 Softwarové vybavení interaktivních tabulí	10
2.2.1 Komerční projekty interaktivních a sdílených tabulí	11
2.2.2 Open Source projekty interaktivních a sdílených tabulí	12
3 Komunikační protokol XMPP	13
3.1 Co je vlastně Jabber/XMPP?	13
3.1.1 Historie	13
3.2 Vlastnosti	14
3.2.1 Shrnutí	16
3.3 Protokol XMPP	16
3.3.1 Model zpráv	16
3.3.2 Síťové spojení	17
3.3.3 Adresování a identity v XMPP	18
3.3.4 Datový stream protokolu XMPP	18
3.3.5 Zprávy protokolu XMPP	18
3.3.6 Zabezpečení protokolu XMPP	21
3.4 Rozšíření	22
4 Sdílení multimediálních dat	23
4.1 Požadované vlastnosti	23
4.2 Síťová architektura	24
4.3 Protokol	24
4.3.1 Přenos textových zpráv	25
4.3.2 Kreslení na tabuli	25

4.3.3	Real time zvuková/obrazová komunikace	26
4.3.4	Ostatní multimediální data	26
4.4	Knihovny pro real-time komunikaci	26
4.4.1	Iris XMPP	26
4.4.2	Telepathy	27
5	Návrh uživatelského rozhraní aplikace	28
5.1	Představení aplikace	28
5.1.1	Název aplikace	28
5.1.2	Využití aplikace	29
5.2	Návrh uživatelského prostředí aplikace	29
5.2.1	Popis návrhu uživatelského rozhraní	30
5.3	Knihovny Qt a prostředí KDE 4	31
5.3.1	Qt 4	31
5.3.2	KDE 4	31
6	Implementace	32
6.1	Návrh aplikace	32
6.1.1	Třída Makneto	32
6.2	Síťová část aplikace	33
6.2.1	Třída Connection	33
6.3	Uživatelské rozhraní	35
6.3.1	Hlavní okno aplikace	36
6.3.2	Postranní panely	36
6.3.3	Panel připojení	36
6.3.4	Panel sezení	36
6.3.5	Panel se seznamem kontaktů	37
6.3.6	Zobrazení přes celou obrazovku	37
6.3.7	Grafika	38
6.4	Kreslení na tabuli	38
6.4.1	Síťová komunikace	38
6.4.2	Příkaz „new“	38
6.4.3	Příkaz „configure“	39
6.4.4	Příklad komunikace	39
6.4.5	Uživatelské rozhraní	40
6.4.6	Kreslicí část okna sezení	41
6.5	Přehrávač multimediálních dat	41
6.5.1	Phonon	42
6.5.2	Přehrávač	43
6.6	Konfigurace aplikace	44
6.7	Plánovaná rozšíření	45
7	Nasazení aplikace	46
7.1	Open Source	46
7.1.1	Licence GNU GPL	46
7.2	Vedení projektu Makneto	47
7.2.1	Správa projektu	47
7.2.2	Repositář zdrojových kódů	47

7.2.3	Prezentace projektu	47
7.3	Nasazení aplikace Makneto	47
7.4	Vydání aplikace	48
7.5	Lokalizace a národní prostředí	48
7.5.1	GNU gettext	48
7.5.2	Překlad aplikace Makneto	49
7.6	Distribuce	49
7.6.1	Specifika cílových platforem	49
7.6.2	Balíčkovací systémy	49
8	Závěr	51
A	The Makneto Handbook	54
A.1	Preface	54
A.1.1	Requirements	54
A.2	Introduction	54
A.2.1	What is Makneto?	54
A.3	Getting started	55
A.3.1	Account settings	55
A.3.2	Go Online!	55
A.4	Using Makneto	55
A.4.1	The Contact List	55
A.4.2	The Shared board	55
A.5	Configuring Makneto	56
A.5.1	The Configure Makneto Dialog	56
A.6	Menu Entries	56
A.6.1	The File Menu	56
A.6.2	The Settings Menu	57
A.6.3	The Help Menu	57
A.7	Credits and Licences	57
B	SPEC soubor	58

Seznam obrázků

2.1	Dotyková obrazovka	9
2.2	SMART Board 660i – Interaktivní bílá tabule pro přední projekci s integrovaným projekčním zařízením a ozvučením	10
3.1	Decentralizace sítě Jabber	14
3.2	Diagram interoperability mezi sítěmi XMPP a ICQ	15
3.3	Model zpráv protokolu XMPP	17
3.4	XML dokument – datový proud obsahující různé typy zpráv	17
3.5	Schéma komunikace Info/Query	20
4.1	Návrh architektury	24
5.1	Diagram případů použití uživatelského rozhraní	29
5.2	Návrh hlavního okna aplikace	30
6.1	Návrh aplikace	32
6.2	Diagram tříd Makneto	33
6.3	Diagram tříd připojení	34
6.4	Diagram tříd rozhraní aplikace	36
6.5	Diagram tříd seznamu kontaktů	37
6.6	Model/View architektura	40
6.7	Diagram tříd kreslení na tabuli	41
6.8	Graf video přehrávače	42
6.9	Diagram tříd videopřehrávače	44
6.10	Diagram tříd datového proudu	44

Kapitola 1

Úvod

1.1 Slovo úvodem

Rozvoj síťových technologií a samotné sítě Internet přináší nové možnosti interaktivní spolupráce a sdílení multimediálního obsahu v reálném čase, a to bez nutnosti nacházet se v jednom okamžiku na stejném místě, v jedné lokalitě, např. v zasedací místnosti u klasické bílé tabule. Naopak je možné, aby se každý jednotlivý účastník zapojil do diskuze z kteréhokoliv místa naší planety, kde je k dispozici dostatečná síťová konektivita. Dochází tak ke snížení nákladů souvisejících s nutností cestování, zajištěním dostatečných prostor a především k úspoře času. Není tak problém přidat se ke svým kolegům, kteří řeší zadaný úkol, z hotelového pokoje na služební cestě a nabídnout tak své poznatky k jeho zdárnému vyřešení nebo se virtuálně sejit s přáteli ze studií v zahraničí, kdy každý sedí u svého počítače ve své zemi.

K takovéto komunikaci se využívá mnoha prostředků. Od čistě textového (a tedy hodně „počítačového“) způsobu, jako jsou takzvané „Instant Messaging“ (IM) služby – velmi populární sítě ICQ, MSN, Jabber atd. – až po sítě určené primárně pro profesionální telekonferenční služby s přenosem hlasu a obrazu diskutujícího účastníka. Časem samozřejmě dochází ke sblížení jednotlivých služeb a tak mnohé IM služby dnes nabízejí přenos hlasu, videa a sdílení souborů, především z důvodu zvýšení pocitu interaktivity účastníků a služby dříve pouze internetové se zapojují do již existujících telekomunikačních sítí a umožňují například volání na klasické veřejné telefonní linky (Skype), či se je přímo snaží nahradit (VoIP, sítě založené na protokolu SIP, ústředny Asterisk atd.). Podobné změny je vidět i ve světě mobilních komunikací, kdy pouhý přenos hlasu je rozšiřován ve 3G sítích o obraz a další multimediální a interaktivní služby.

1.2 Historie počítačové komunikace v reálném čase

První pokusy se zasíláním krátkých zpráv v reálném čase se datují do doby vzniku prvních víceuživatelských systémů pro komunikaci připojených uživatelů pomocí terminálů. S rozšiřováním sítí vznikaly další většinou peer-to-peer protokoly jako klasický unixový talk a klient server systémy jako IRC. Ke skutečnému rozšíření real time komunikace došlo ovšem až s rozšířením připojení běžných uživatelů do sítě Internet v první půli devadesátých let minulého století. První vlaštovkou bylo ICQ v roce 1996, následováno AOL Instant Messengerem (rok 1997). Další velké společnosti nezháhaly a vznikla spousta vzájemně nekompatibilních protokolů pro Instant Messaging.

Z VoIP aplikací stojí za zmínku dnes již nepodporovaný NetMeeting od společnosti Microsoft, který nabízel real time sdílenou tabuli, videokonference a další služby jako sdílení souborů. Malou revoluci způsobila teprve nedávno aplikace Skype uvedená v roce 2003, která k hlasové komunikaci přes Internet přilákala běžné uživatele počítačů.

1.3 Současnost

V současné době se situace na trhu s IM službami ustálila tak, že mezi světově nejrozšířenější služby patří Jabber (využívaný hojně ve formě vlastních interních sítí) s cca 40-50 milióny uživatelů, MSN (dnes Windows Live Messenger) s cca 28 milionů aktivních uživatelů a AOL IM a v České republice především ICQ (pět milionů aktivních uživatelů v samotné Evropě). V hlasových real time komunikačních službách vládne Skype a služby na bázi protokolu SIP, obojí nabízející i propojení do běžných telefonních sítí. Přesné porovnání podle počtu uživatelů není možné, jedná se o odhady, především v decentralizovaných sítích, jako je Jabber, je obtížné spočítat všechny uživatele, u komerčních služeb zase vyšší číslo působí lépe marketingově a může být z důvodů prodeje nadsazené. Nadále dochází ke slučování služeb do aplikací typu „vše v jednom“, takže například Windows Live Messenger se nedá již brát jako čistě IM klient, ale jedná se o aplikaci nabízející textové zprávy, přenos obrazu a zvuku, hraní atd., podobně se rozšiřují i další služby jako ICQ a Skype.

1.4 Cíl projektu

Cílem projektu je navrhnout aplikaci, která umožní spolupráci více uživatelů pomocí interaktivní multimediální tabule. Účastníci takového sezení u této virtuální tabule budou mít pocit jako by byli doopravdy společně přítomni v nějaké jednacím místnosti. Aplikace jim umožní textovou komunikaci ve stylu instant messagingu, kresbu na tabuli, takže budou moci ostatním účastníkům jednoduše a přirozeně v grafickém provedení prezentovat své myšlenky a nápady, hlasovou a obrazovou komunikaci. Tato virtuální tabule bude navíc nabízet i funkce, které u jejího vzoru z reálného světa nejsou běžně možné a to např. přehrávání videí, interaktivní aplikace apod.

1.5 Obsah technické zprávy

V kapitole „Interaktivní tabule“ se čtenář seznámí s prostředky pro styk počítač-člověk, které slouží ke kreslení a prezentaci jako na klasické bílé tabuli. Budou probírány jak základní zařízení jako tablet, tak hlavně specializované systémy tzv. interaktivních tabulí. Bude nastíněno jejich využití v oblasti interaktivní výuky a interaktivních učeben.

V následující kapitole se čtenář dozví podrobné informace o protokolu XMPP, tedy o „Extensible Messaging and Presence Protocol“. Jedná se o otevřený protokol pro zasílání zpráv a prezenci v síťovém prostředí.

Poté je v kapitole „Sdílení multimediálních dat“ popsáno využití probraného protokolu XMPP v rámci projektu – tedy jako protokolu přenášejícího jak samotné kreslení po tabuli, tak jeho využití v přenosu multimediálních dat a při přenosu zvuku a obrazu. Pro účely aplikace zde bude navržen formát zpráv komunikace protokolu XMPP.

Neméně důležitým aspektem vzniku kvalitní aplikace je návrh jejího uživatelského rozhraní. Čtenář se v kapitole „Návrh uživatelského rozhraní“ podrobně dozví o jednotlivých funkčních částech grafického rozhraní aplikace a jejich využití uživatelem pomocí případů

použití včetně jejich předběžných náčrtů, podle kterých bude postupováno při návrhu skutečné aplikace v prostředí toolkitu Qt 4.

Kapitola „Implementace“ je popsána v samostatné části práce. Zde jsou shrnuty postupy a výsledky práce na aplikaci. V této kapitole je naznačen i budoucí vývoj aplikace.

Kapitola „Vedení Open Source projektu a nasazení“ popisuje proces správy projektu využitý při realizaci, od samotného vývoje až po distribuci k finálnímu uživateli.

V závěrečné kapitole je učiněno ohlédnutí za celou prací a její finální zhodnocení včetně zhodnocení přínosu aplikace v oblasti vývoje open source projektů. Je zároveň prodiskutován další budoucí vývoj aplikace.

1.5.1 Semestrální projekt

Diplomová práce navazuje na semestrální projekt, ve kterém byly řešeny první tři body zadání, odpovídající kapitolám Úvod, Interaktivní tabule, Komunikační protokol XMPP, Sdílení multimediálních dat a Návrh uživatelského rozhraní.

Vývoj projektu probíhá podle zadání firmy Red Hat a ve spolupráci s jejich programátory. Společnost se zabývá vývojem open source aplikací s pobočkami a zaměstnanci v různých částech světa a tento projekt má sloužit pro zvýšení efektivity komunikace.

Kapitola 2

Interaktivní tabule

Interaktivní tabule jsou systémy, které usnadňují interakci mezi počítačem, přednášejícím a jeho posluchači. Jejich využití je velmi široké – zahrnuje:

- interaktivní učebny pro školy,
- přednáškové sály či
- jednací místnosti.

V této kapitole jsou popsány technické prostředky pro komunikaci mezi uživatelem a počítačem, především s přihlédnutím ke komunikaci pomocí interaktivních tabulí.

2.1 Technické vybavení pro interakci s počítačem

2.1.1 Běžná polohovací zařízení

Mezi běžná zařízení, které je možné použít pro kreslení na obrazovce patří např. počítačová myš a touchpad. Ty ovšem nenabízí požadovanou přesnost a především snadnou použitelnost a omezují interaktivitu mezi počítačem a uživatelem.

2.1.2 Tablety

Tablety jsou zařízení, které uživateli dovolují ovládat počítač pomocí kresby tužkou (stylusem) na speciální vrstvě citlivé na dotek. Jsou vhodné pro ovládání grafických aplikací a oproti běžným polohovacím zařízením zvětšují kvalitu i požitek z kresby. Přesto stále postrádají vhodný druh interaktivity vyžadovaný pro interaktivní tabule, jelikož uživatel pod tužkou nevidí skutečný výsledek své práce, ale musí ho sledovat na zobrazovacím zařízení, které může například v případě projekce být i za jeho zády. V prostředí interaktivních tabulí a interaktivních výukových místnostech může být velmi vhodný jako komunikační doplněk k hlavní tabuli a dá se využít pro interakci ze strany posluchačů.

2.1.3 Dotykové obrazovky

Dotykové obrazovky rozšiřují možnosti tabletů o vizuální informaci – uživatel zde vidí svoji práci v místě, ve kterém se pohybuje virtuální tužkou po obrazovce. Výsledný dojem je tedy mnohem skutečnější a dovoluje o to větší soustředění na vlastní ovládání počítače. Například kresba na obrazovce se poté neliší od kresby skutečnou tužkou na papír. Ovládání

dotykových obrazovek je možné buď dotykem prstů nebo perem (nazývaným také stylus) jak je možné vidět na ilustračním obrázku 2.1.

Technicky jsou dotykové obrazovky sestaveny z běžné obrazovky (může být CRT, tak i LCD), dotykového senzoru a jeho řadiče, který komunikuje s PC (v dnešní době běžně přes rozhraní USB, dříve například přes sériový port, u specializovaných zařízení přímo s procesorem přes jeho vstupní porty).

Nejčastěji dotykový senzor bývá čirý skleněný panel s vrstvou reagující na dotek, který svou velikostí odpovídá velikosti zobrazované plochy na obrazovce a připojuje se nad ni. Existuje mnoho technologií použitelných pro konstrukci dotykové vrstvy – resistivní, akustické, kapacitní, infračervené atd.

Použití dotykových obrazovek v interaktivní komunikaci s počítačem je vzhledem k jejich velikosti výhodné hlavně pro osobní použití. Ve vztahu k této diplomové práci a zde navrhované aplikaci se jedná dle mého názoru o nevhodnější prostředek, protože primární zaměření je na komunikaci přes síť a není potřeba prezentace pro více lidí v rámci místnosti, či přednáškového sálu. Ovšem ani toto využití se samozřejmě nevylučuje, viz dále, a je přes projekční zařízení možná prezentace na dotykovém displeji i širšímu publiku. S rozvojem technologie LCD navíc není problém vyrábět dotykové displeje větších velikostí a přiblížit se tak specializovaným interaktivním tabulím (např. systém SMART Board pro plazmové/LCD televizory).

Dotykové obrazovky mají samozřejmě mnohem větší praktické využití a setkáváme se s nimi stále častěji – v mobilních telefonech, přehrávačích hudby, PDA, navigacích i např. v obchodech na váhách v oddělení ovoce a zeleniny k výběru druhu zboží a k tisku jeho etikety s čárovým kódem.



Obrázek 2.1: Dotyková obrazovka

2.1.4 Specializované systémy interaktivních tabulí

Specializované systémy interaktivních tabulí sestávají většinou ze spojení běžného projektoru a zařízení schopného zachytit pohyb ukazovátka po projekční ploše. Na tabuli je promítán reálný obraz z počítače, který je pak možné ovládat rukou, ukazovátkem či psát

běžnou stíratelnou fixou s tím, že text je zaznamenán v počítači. V závislosti na snímacím systému je možné využívat jen základní ovládání nebo kombinaci všech výše uvedených způsobů.

Interaktivní tabule tedy spojuje výhody, projekční plochy, bílé propisovací tabule a dotykové obrazovky v jednom zařízení. Na obrázku 2.2. je tato tabule znázorněna. Skládá se ze dvou částí – samotného plátna a zařízení, které promítá obraz a sleduje pohyb ukazovátka. Podle druhu tabule (viz dále) se může jednat o jeden celek, případně více zařízení (např. v případě využití již instalovaného projektoru).

Využití interaktivních tabulí a ploch je především pro prezentační činnost pro více posluchačů. V tomhle ohledu těží především z velkého obrazu díky využití projektorů, případně plazmových či LCD televizorů s velkou úhlopříčkou.

Na českém trhu se lze setkat s nabídkou mnoha světových výrobců, nejčastěji se jedná o systémy SMART od kanadské firmy Smart Technologies.

V nabídce jsou systémy

- SMART Board pro přední projekci
- SMART Board pro zadní projekci
- SMART Board pro plazmové displeje



Obrázek 2.2: SMART Board 660i – Interaktivní bílá tabule pro přední projekci s integrovaným projekčním zařízením a ozvučením

2.2 Softwarové vybavení interaktivních tabulí

Neméně důležitou součástí interaktivních systémů, jako jsou interaktivní tabule, je jejich softwarové vybavení. Jednotlivý software se liší především v hlavním zaměření – některé

systemy jsou určeny přímo pro interaktivní tabule a nabízí slabší možnosti použití jako sdílené tabule, některé systémy jsou zaměřeny spíše pro výukové potřeby do učeben i pro základní školy a tedy vyžadují jednoduché prostředí, které tak ale postrádá některé rysy pro obecné nasazení. Dalším druhem jsou softwarové aplikace zaměřující se spíše na komunikaci po síti přes běžná kancelářská a domácí PC. Výhoda je tedy v síťovém použití, nevýhoda v méně snadném použití pro větší systémy interaktivních tabulí (především v návrhu uživatelského prostředí určeného pro jednoho uživatele na monitoru PC).

2.2.1 Komerční projekty interaktivních a sdílených tabulí

V následujícím přehledu jsou tedy diskutovány systémy z každé výše uvedené kategorie, i když v mnohých případech dochází k jistému překrytí nabízených vlastností.

SMART Board software

SMART Board software je programové vybavení k interaktivním tabulím společnosti SMART Technologies představeným výše. Instalace obsahuje balík nástrojů pro ovládání tabulí (ovladače, firmware), kreslicí plochu nazývanou Notebook a další nástroje pro ovládání tabule.

Aplikace Notebook je zaměřena především pro prezentace a k ovládání interaktivních tabulí, síťová podpora je ve formě sdílení vytvořených galerií v síti.

SMART Board software nabízí:

- podporu interaktivních tabulí společnosti SMART Technologies,
- nástroje pro přípravu prezentací,
- předpřipravené galerie klipartů pro využití v prezentaci a
- přidávání příloh k prezentaci.

SMART Board software je dodáván zdarma pro operační systémy Windows, Linux a Mac OS, ke stažení z domovské stránky společnosti. Licence ovšem zakazuje použití na jiných tabulích než od společnosti SMART. Výjimkou je pouze příprava prezentací a materiálů vyučujícími na svých počítačích.

NotateIt

NotateIt od společnosti Blade Software je zaměřením nejbližší představě cíle tohoto diplomového projektu. Dle výrobce produkt simuluje vícestránkovou magnetickou tabuli na PC a dělá ho tím jednodušší pro použití k brainstormingu, myšlenkovým mapám (mind maps), ke sbírání, organizování a ukládání myšlenek a ideí [9].

NotateIt nabízí anotace obrazovek spuštěných aplikací, nahrávání kresby na tabuli, přehrávač videí, softwarovou klávesnici, vkládání obrázků a animací na tabuli, změna vzhledu tabule a podporu interaktivních tabulí, tablet PC a zařízení používaných v této oblasti.

Síťová komunikace aplikace však není řešena a omezuje tedy výrazně její použití pro osobní použití a tvorbu prezentací. Velmi dobrá je také podpora interaktivních tabulí. Podstatnou nevýhodou je omezení pouze na platformu MS Windows. Jedná se o komerční

aplikaci v cenách od 25 euro a 45 euro podle druhu licence v závislosti na její přenositelnosti na více počítačů.

2.2.2 Open Source projekty interaktivních a sdílených tabulí

Software interaktivních tabulí bývá většinou jejich přímou součástí a tak na se na poli open source projektů dá najít spíše interaktivní sdílení obsahu, než vybavení pro samotné tabule. Většinou se jedná o rozšíření již existujících aplikací o možnosti sdílení kreseb na tabuli.

Inkscape

Zástupcem sdílení kreslení je známý vektorový grafický program Inkscape, který v současné době nabízí možnost sdílení samotného kreslicího plátna ve formě interaktivní tabule. Grafici mohou spolupracovat na kresbě v reálném čase. Software je tedy převážně zaměřen na sdílené spolupráci v oblasti vektorové grafiky, ale díky velké výbavě Inkscape různými grafickými nástroji není problém využít jako sdílené tabule i pro jiné účely.

Kapitola 3

Komunikační protokol XMPP

XMPP [10] – „Extensible Messaging and Presence Protocol“ je otevřený protokol založený na XML určený pro komunikaci pomocí zpráv (Instant Messaging, dále zkráceně IM) a informace o stavu dostupnosti uživatelů. Jedná se o základní protokol systému Jabber. Je standardizován organizací IETF a vyvíjen v rámci komunity od roku 1999. Navrhovaná aplikace je založena na protokolu XMPP a proto je mu věnována celá tato kapitola s jeho představením a hlubším popisem.

3.1 Co je vlastně Jabber/XMPP?

Odpověď na tuto otázku záleží na úhlu pohledu tazatele. Na Jabber/XMPP se lze totiž dívat z mnoha pohledů[1]:

- Jabber/XMPP je technologie
- Jabber/XMPP je protokol
- Jabber je Instant Messaging systém založený na XML
- Jabber je implementací souboru protokolů

3.1.1 Historie

V lednu 1999 Jeremie Miller oznámil existenci Jabberu a během tohoto roku proběhl bouřlivý vývoj – jak na poli serveru (jabberd), tak na straně klientů a komunikačních knihoven – a již téhož roku byla navržena standardizace organizací IETF. V roce 2000 došlo k prvním pokusům o standardizaci ve skupině Instant Messaging and Presence Protocol (IMPP) Working Group, ovšem vzhledem k neshodám IMPP WG nevydala žádný protokol. Proto byla v říjnu 2001 zformována organizace Jabber Software Foundation za účelem koordinace dalšího vývoje a standardizace technologií postavených na Jabberu.

Ovšem teprve začátkem roku 2002 byl znovu poslán další návrh do IETF, který byl schválen jako tzv. Internet-Draft a tento úspěch povzbudil komunitu k pokusu o opětovné prosazení IETF Working Group a formalizaci protokolu Jabber pod neutrálním názvem Extensible Messaging and Presence Protocol (XMPP). Výsledkem byly dva hlavní Internet-Drafts pro „XMPP Core“ a „XMPP IM“. V roce 2004 byla dokončena standardizace protokolu jako Proposed Standard. Výsledkem jsou následující dokumenty RFC:

- RFC 3920 – Extensible Messaging and Presence Protocol (XMPP): Core,
- RFC 3921 – Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence,
- RFC 3922 – Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM),
- RFC 3923 – End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP).

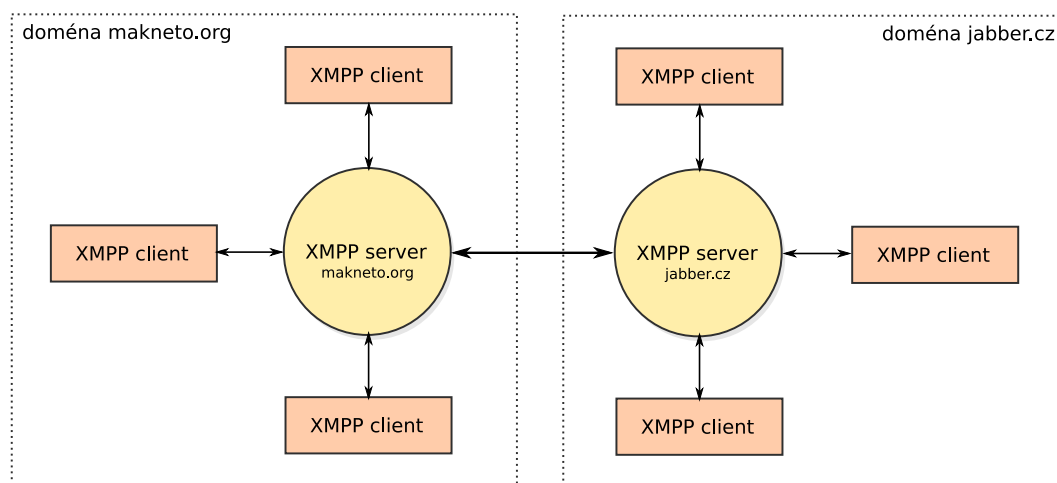
Další vývoj pak směřoval a nadále směřuje k tvorbě různých rozšíření protokolu, případně k opravám RFC 3920 a RFC 3921 a cílové snaze o rozšíření na Draft Standard.

Začátkem tohoto roku došlo k přejmenování Jabber Software Foundation na XMPP Standards Foundation a k dalšímu zaměření organizace na vývoj standardu XMPP místo vývoje samotného software.

3.2 Vlastnosti

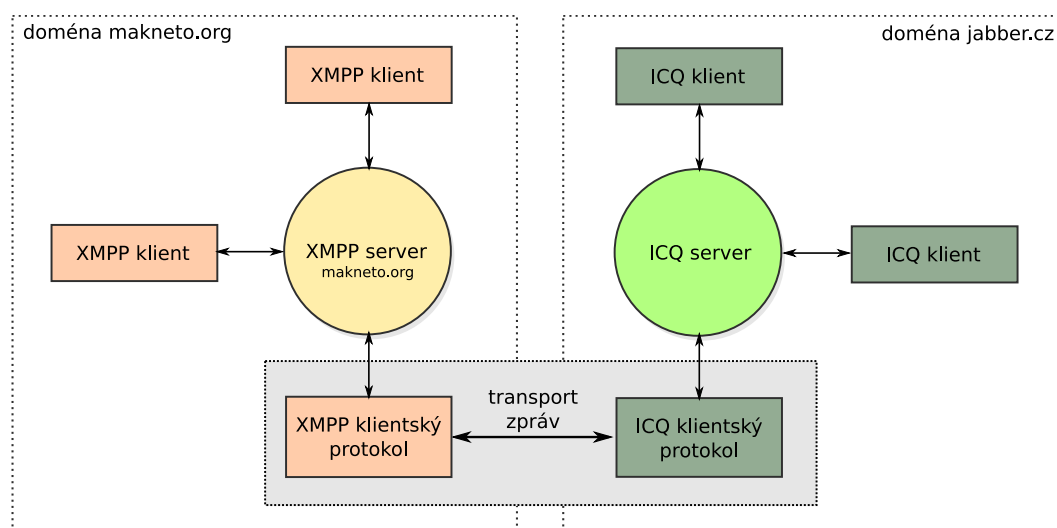
Cílem mnohaletého vývoje protokolu XMPP byla snaha vytvořit co možná nejlepší IM systém s podporou real time sledování prezenze a posílání zpráv [8]. Pro dosažení tohoto nelehkého úkolu bylo potřeba navrhnout systém o vhodných vlastnostech tak, aby bylo možné říct, že došlo ke zlepšení stávajícího stavu. Mezi zmíněné vlastnosti patří:

- *Decentralizace.* Architektura sítě XMPP je decentralizovaná, neexistuje žádný centrální server, jehož výpadek znamená výpadek celé sítě. Svojí architekturou připomíná elektronickou poštu - každý může mít vlastní server, poskytující služby svým klientům a tyto servery spolu mohou navzájem komunikovat, jak je ilustrováno na obr. 3.1.



Obrázek 3.1: Decentralizace sítě Jabber

- *Úplně otevřený systém a standard.* Na rozdíl od ostatních IM systémů, je XMPP (a vždy bude), kompletně otevřený systém, který tak dovoluje každému možnost vlastní implementace ve vlastní aplikaci/systému bez nutnosti jakékoliv platby či dalších omezujících podmínek. Zároveň dává možnost podílet se na vlastním vývoji protokolu a jeho rozšiřování.
- *Založený na XML.* Extensible Markup Language je v současné době prakticky standard pro výměnu informací, kde je zapotřebí dosáhnout maximální otevřenosti okolnímu světu. Díky všeobecné znalosti je XML velmi vhodnou volbou, především s ohledem na budoucí vývoj protokolu.
- *Flexibilita.* Protokol XMPP je velmi flexibilní a rozšiřitelný. Je možné vyvíjet vlastní rozšiřující části systému. Nejzajímavější rozšíření oficiálně spravuje XMPP Standards Foundation.
- *Interoperabilita.* Systém umožňuje propojení jak v rámci sítí založených na protokolu Jabber, viz. decentralizace, tak i propojení s ostatními IM systémy (a nejen s IM systémy). Pro připojení k více druhům služeb lze tedy využít jednoho protokolu – XMPP a tím i využít všech ostatních výhod plynoucích ze zde vyjmenovaných vlastností. Na obr. 3.2 je znázorněné propojení se sítí ICQ pomocí tzv. transportu zpráv. Stejným způsobem je možné dosáhnout propojení s libovolnou IM sítí.



Obrázek 3.2: Diagram interoperability mezi sítěmi XMPP a ICQ

- *Bezpečnost.* XMPP servery mohou běžet v rámci vnitřní sítě, například na firemním intranetu bez nutnosti komunikovat s okolním světem. Při komunikaci po nezabezpečené síti je možné využít šifrovaného přenosového kanálu a ověřování identity uživatelů pomocí certifikátů. Díky otevřenosti standardu se není potřeba obávat tajných zákoutí a bezpečnostních děr jako u proprietárních systémů.
- *Kontrola nad systémem.* Žádná organizace či skupina nemá neomezenou moc nad systémem jako takovým. Vývoj je veden pod taktovkou XMPP Standards Foundation a kdokoliv má možnost předkládat nové vylepšení, které jsou schvalovány komunitou uživatelů.

3.2.1 Shrnutí

Protokol XMPP je využíván díky své otevřenosti v řadě aplikací různého poslání. Nejznámější využití je pro komunikaci pomocí zpráv (IM). Dále se často využívá jako protokol pro sjednávání komunikace pro výměnu multimediálních dat (např. VoIP), sdílené tabule a spolupráci v rámci kolaborativního prostředí. Všechny zmíněné využití odpovídají záměru této diplomové práce a díky všem vlastnostem se protokol XMPP stal jednoznačnou volbou pro její zpracování.

3.3 Protokol XMPP

Hlavní součásti protokolu XMPP jsou detailně popsány v dokumentech RFC 3920 a RFC 3921, které jsou veřejně dostupné na webových stránkách organizace XMPP Standards Foundation. Nemá tedy smysl opisovat je zde do posledního detailu a tato podkapitola bude spíše výtahem nejdůležitějších vlastností protokolu tak, aby čtenář rozuměl jeho principům a po prostudování byl schopen navrhnout a implementovat systém založený na protokolu XMPP.

XMPP je souhrn několika základních protokolů a to především:

- Extensible Messaging and Presence Protocol (XMPP): Core a
- Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence.

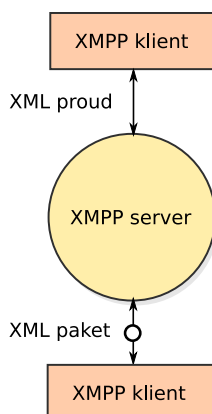
Protokol Core tvoří základ komunikace mezi klientem a serverem (případně servery) – definuje připojení a výměnu obecných zpráv. Druhý jmenovaný protokol Instant Messaging and Presence popisuje začlenění XMPP do prostředí IM sítí – tedy prezenci uživatelů a posílání IM zpráv.

3.3.1 Model zpráv

Model zpráv protokolu XMPP je obecně založený na architektuře klient-server (která ovšem nemusí být výlučná a je možné využít i jiných modelů jako peer to peer apod.). Model je naznačen na obr. 3.3 a ukazuje dva komunikující klienty prostřednictvím XMPP serveru.

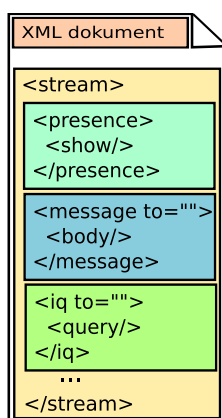
Hlavními prvky tohoto modelu jsou:

- *Server*. Primární odpovědností XMPP serveru je nabízet komunikační služby protokolu klientům. Hlavní službou je směrování zpráv v síti uživatelům a správa jejich uživatelských účtů. Vzhledem k otevřenosti protokolu je možné další rozšíření nabízených služeb.
- *Klient*. Klientem se rozumí uživatelská aplikace zobrazující příchozí zprávy, prezenci a seznam uživatelů. Klient může působit i zcela autonomně na uživateli a chovat se tedy jako tzv. chatbot.
- *Proud*. (*stream*) Síťové spojení mezi klientem a serverem je realizováno pomocí páru jednosměrných datových *XML proudů*. Z pohledu XML se jedná a proud dokumentu uzavřeného mezi tagy `<stream>` a `</stream>`.



Obrázek 3.3: Model zpráv protokolu XMPP

- *Pakety*. (*stanzas*) XML fragmenty posílané v proudu mezi klientem a serverem se nazývají *pakety*. Každý paket je sebeobsahující, validní XML poddokument. XMPP protokol definuje jejich formát a způsob jejich zpracování.



Obrázek 3.4: XML dokument – datový proud obsahující různé typy zpráv

Na obr. 3.4 je znázorněna struktura XML dokumentu v prostředí protokolu XMPP. Dokument je zabalen v kořenovém prvku datového proudu a obsahuje všechny tři typy možných zpráv protokolu XMPP.

3.3.2 Síťové spojení

Síťové spojení je v rámci XMPP realizováno pomocí protokolu TCP/IP. Jabber server obvykle běží na portu 5222 pro nešifrované spojení, 5223 pro SSL šifrované spojení. Je ale také možné využít tunelovaného přenosu přes protokol HTTP na portu 80. Toho se hojně využívá v sítích, ve kterých jsou jiné než běžné protokoly zakázány ve firewallu.

3.3.3 Adresování a identity v XMPP

Jabber ID

Základem pro adresování a identifikaci všech entit v protokolu XMPP je tzv. *Jabber ID*, zkráceně JID. Svým významem pro Jabber odpovídá Uniform Resource Locatoru (URL) známému z protokolu HTTP (HyperText Transfer Protocol) a v některých zápisech se blíží podobě emailové adresy (v mnohých službách, např. v GTalku, může JID sloužit jako obojí).

Tvar JID je [*node* @] *domain* [, / ' *resource*], kde:

- *Node* je nepovinný parametr identifikující entitu připojující se k serveru, tedy například uživatelské jméno atd.
- *Domain* doména nebo hostname.
- *Resource* je druhý nepovinný parametr, používaný pro kvalifikaci dotazů.

3.3.4 Datový stream protokolu XMPP

Atributy datového proudu:

- *to* – adresát
- *from* – odesílatel
- *id* – identifikace
- *xml:lang* – jazyk
- *version* – verze

3.3.5 Zprávy protokolu XMPP

Protokol XMPP definuje následující obsah paketů:

- *Zprávu* (*message*)
- *Informaci* o prezenci (*presence*)
- *Info/dotaz* (*Info/Query*)

Zpráva (*message*) je nejdůležitějším a nejpoužívanějším paketem v síti protokolu XMPP. Prakticky většina komunikace probíhá právě pomocí zpráv. Informace o prezenci (*presence*) je speciální druh paketu pro zjišťování aktivity uživatelů. Následně pak *Info/dotaz* (*Info/Query*) je obdoba požadavku GET v protokolu HTTP.

Zpráva

Typy zpráv (definovány v atributu *type*) tagu <message>, protokol XMPP podporuje šest druhů zpráv:

- *normal* – běžná zpráva (jako elektronická pošta)
- *chat* – rozhovor uživatel s uživatelem

- *groupchat* – rozhovor v místnosti
- *headline* – titulní zpráva
- *error* – chybová zpráva
- *x:oob* – out-of-band zpráva pro výměnu informací ve styku klient s klientem

Běžný klient pak implementuje následující podpakety paketu `<message>`:

- `subject` – Obsahuje předmět zprávy (jako v elektronické poště).
- `thread` – Identifikace zpráv v rámci tzv. vlákna.
- `body` – Obsah samotné zprávy.
- `error` – Standardní chybová zpráva.

Zpráva pak může vypadat následovně:

```
<message from=„rezza@jabber.cz/skola“
to=„kamarad@jabber.cz/doma“ id=„1“>
  <thread>vlakno_1</thread>
  <subject>Pozdrav</subject>
  <body>Zdravím domů ze školy v rámci těla zprávy.</body>
</message>
```

Ta obsahuje zprávu od uživatele *rezza@jabber.cz* připojeného ve škole pro uživatele *kamarad@jabber.cz*, který se pravděpodobně podle svého zdrojového popisu nachází doma. Identifikátor se většinou neuvádí. Pomocí paketu `<thread>` jsou udržovány při sobě zprávy v rámci jednoho vlákna a označeny identifikátorem vlákna `vlakno_1`. Implicitně se předpokládá typ zprávy *normal*.

Ve chvíli, kdy server přijme takovou zprávu, pokusí se ji doručit adresátovi. Pokud je adresát nepřipojen, je zpráva uložena na serveru do chvíle než se uživatel připojí a server mu zprávu doručí.

Prezence

Instant messaging služby se odlišují od běžného emailu především interaktivitou komunikace a to díky možnosti zjistit, jestli je či není vybraný uživatel připojený (či v nějakém jiném stavu jako nedostupný apod.).

Prezence se v rámci protokolu XMPP řeší pomocí paketu `<presence>`, který obsahuje standardní atributy jako zpráva, tedy `to`, `from` a `type`. Poslední jmenovaný atribut odpovídá typu prezence (*available*, *unavailable* atd.).

Tag `<presence>` může obsahovat další podelementy, a to:

- `status` – textový popis stavu (například „Nerušit, studuji“).
- `priority` – priorita daného zdroje, identifikována nezáporným číslem. Zprávy jsou směrovány zdroji, který má nejvyšší prioritu a je dostupný (*available*).

- error – standardní chybový paket.
- show – stav dostupného klienta. Standardně je klient ve stavu `normal/online` – tedy dostupný a přijímající zprávy běžným způsobem. Další stavy jsou:
 - chat – uživatel vyhledává aktivně rozhovor.
 - away – uživatel je na krátkou dobu pryč.
 - xa – uživatel je pryč po delší dobu.
 - dnd – uživatel si nepřeje být vyrušován.

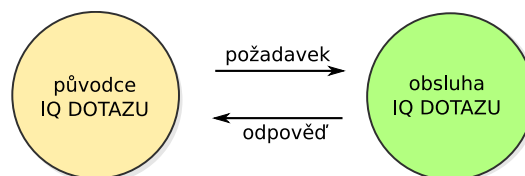
Prezence pak může vypadat následovně:

```
<presence from=„rezza@jabber.cz/skola“ to=„jabber.cz“
type=„available“>
  <status>Právě píšiš diplomku</status>
  <priority>Pozdrav</priority>
  <show>dnd</show>
</presence>
```

Nejjednodušší paket může být čistě jen `<presence/>` oznamující, že uživatel je k dispozici. Uživatelé, kteří mají zájem získávat informaci o prezenci vybraného uživatele se přihlásí k odebrání prezence na serveru a ten je pak automaticky obesílá s každou změnou. Využívá se atributu `type` s parametrem `subscribe` a server odpoví odpovědí s parametrem `subscribed`. Odhlašování se pak děje klasicky opačnými parametry `unsubscribe` a odpovědí `unsubscribed`.

Info/Query

Info/Query (zkráceně IQ) jsou pakety umožňující komunikaci stylu požadavek-odpověď a rozšiřují tak možnosti statického způsobu zaslání zpráv. Díky IQ je protokol XMPP velmi přizpůsobitelný a rozšiřitelný. IQ pakety jsou totiž hlavně základ pro další rozšíření protokolu XMPP, využívají hojně jmenných prostorů ve standardu XML a díky tomu je možné vytvářet validní XML poddokument se zachováním validity celého dokumentu XML streamu.



Obrázek 3.5: Schéma komunikace Info/Query

Paket IQ je specifikován tagem `<iq>`, který má opět atributy `to`, `from`, v mírně jiném významu a standardně `id` a `type`. Parametry atributu `type` jsou `set` a `get`, které se posílají obsluze (handleru) dotazu. Obsluha v případě úspěchu odešle paket s parametrem `result` atributu `type` nebo `error`.

Dotaz pak může vypadat následovně:


```

<iq type=„get“ to=„slovník@jabber.cz“ from=„rezza@jabber.cz“>
  <query xmlns=„my:dictionary:namespace“>
    <word>Hi</word>
    <dir>en-cz</dir>
  </query>
</iq>

```

IQ pakety se používají například při registraci uživatele. Nejdřív se pošle `probe` paket typu `get`, k otestování, zda server umožňuje registraci a pokud ano, klient odešle paket `register` typu `set`.

3.3.6 Zabezpečení protokolu XMPP

V XMPP protokolu je zabezpečení přímo součástí specifikace a oproti jiným systémům, jak už IM, tak například elektronické poště, je právě na bezpečnost kladem velký důraz.

Identita

Identita uživatele je v XMPP ověřována a tudíž není možné vydávat se za jiného uživatele bez získání hesla například podvržením paketů zprávy. Spolu s autorizací kontaktů a seznamy povolených uživatelů (whitelist) se jedná o velmi dobrou ochranu proti nevyžádaným zprávám. Ověření identity může být dále posíleno autentizací přes certifikáty.

Šifrování datových proudů

Samotný datový proud v protokolu XMPP může být šifrován pomocí protokolu TLS – Transport Layer Security. Jmenný prostor pro TLS je „urn:ietf:params:xml:ns:xmpp-tls“. K zahájení TLS komunikace se využívá elementu `<starttls>` a domluva o TLS musí být provedena ještě před započítím SASL autentizace.

Pokud server požaduje TLS, započne komunikaci následovně:

```

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

```

Klient odpoví:

```

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

```

Po domluvě na TLS klient zahájí nový datový proud. Následně může pokračovat komunikace SASL autentizací.

Autentizace uživatelů

Autentizace uživatelů je popsána přímo v RFC 3920, tedy tzv. Core části specifikace protokolu, a existuje několik přístupů, které se liší úrovní zabezpečení – od prostého nešifrovaného hesla, s nejnižší (žádnou!) úrovní zabezpečení, až po využití prostředí SASL – Simple Authentication and Security Layer, který je prakticky standardem pro autorizaci v mnoha

protokolech ve spojení s SSL a ověřením pomocí certifikátů.

Původní specifikace protokolu Jabber/XMPP používala k autentizaci uživatelů vlastní protokol definovaný ve jmenném prostoru „jabber:iq:auth“, který byl následně při formalizaci organizací IETF nahrazen protokolem SASL (který je definován v RFC 4422). SASL nabízí flexibilnější přístup k autentizaci s využitím všech jeho metod (PLAIN, DIGEST-MD5, EXTERNAL, ANONYMOUS). Tento původní protokol je možné nalézt v dokumentu XEP-0078: Non-SASL Authentication z důvodu zpětné kompatibility a pomalejšího přechodu na nový protokol.

Autentizace probíhá podle následujícího postupu. Nejdříve je navázáno klasicky spojení mezi klientem a serverem a je vytvořen datový proud (`<stream::stream>`), v tuto chvíli není ještě vytvořené sezení (session) a případně doručené zprávy jsou ukládány do fronty. Server zašle klientovi seznam tzv. „features“, tedy událostí, kterým rozumí. Ty obsahují přihlašovací mechanismy v jmenném prostoru „urn:ietf:params:xml:ns:xmpp-sasl“. Nejčastěji se v rámci protokolu XMPP (a taky většina XMPP serverů implementuje) mechanismy PLAIN a DIGEST-MD5 (zde je změna oproti původní specifikaci využívající pro DIGEST algoritmu SHA-1).

```
<stream:features>
  <mechanisms xmlns='urn:ietf:params:xml:ns:xmpp-sasl'>
    <mechanism>DIGEST-MD5</mechanism>
    <mechanism>PLAIN</mechanism>
  </mechanisms>
</stream:features>
```

Klient si následně vybere požadovaný mechanismus a provede se autentizace podle daného RFC SASL. Pro nejčastěji používaný mechanismus DIGEST-MD5 server zašle BASE64 kódovanou výzvu (challenge) a klient na tuto výzvu odpoví (response). Pokud je autentizace úspěšná, klient započne nový datový proud se serverem. V tomto datovém proudu již je možné navázat sezení.

3.4 Rozšíření

Nejvýraznější vlastností protokolu XMPP je jeho rozšiřitelnost.

Vzhledem k projektu jsou užitečné následující rozšíření:

- XEP-0045: Multi-User Chat (MUC), tedy komunikace více uživatelů v rámci jedné místnosti,
- XEP-0096: File Transfer (pro přenos souborů),
- XEP-0166: Jingle (viz dále),
- XEP-0167: Jingle Audio via RTP,
- XEP-0180: Jingle Video over RTP.

Zajímavé rozšíření je Jingle – používá se k navázání peer to peer spojení a vytvoření spojení, které je určené především pro přenos hlasu a videa. Ty jsou definovány v Jingle Audio over RTP a Jingle Video over RTP.

Kapitola 4

Sdílení multimediálních dat

Základní problém návrhu síťové aplikace je vymyšlení vhodného způsobu, jakým bude aplikace komunikovat s okolním světem. Cílem je navržení protokolu umožňujícího komunikace v reálném čase s různými druhy multimediálních i nemultimediálních dat.

Pro potřeby projektu je potřeba vyřešit několik základních problémů. Je to především navázání spojení do sítě služby, aktuální stav prezenze uživatelů, vytvoření sezení pro sdílení tabule. Součástí je také klasické zasílání instantních textových zpráv, jak je známé z běžných IM aplikací. Důkladnou diskuzi si zaslouží především problém přenosu dat na sdílené tabuli – jak samotného kreslení, tak doplňujících objektů, jako jsou obrázky a videa. Zcela vlastní a samostatnou kapitolu komunikace tvoří přenos hlasových hovorů a videokonferencí.

V této kapitole bude rozebráno jak vhodně navrhnout protokol tak, aby bylo možné v co největší míře využít již existujících standardů (např. v předchozí kapitole popsany protokol XMPP). To je velmi důležité vzhledem k interoperabilitě s řešením stejného druhu komunikace třetích stran. Navrhovaná aplikace v sobě kombinuje mnoho rozličných přístupů a to přináší zvýšené požadavky na návrh i konečnou implementaci.

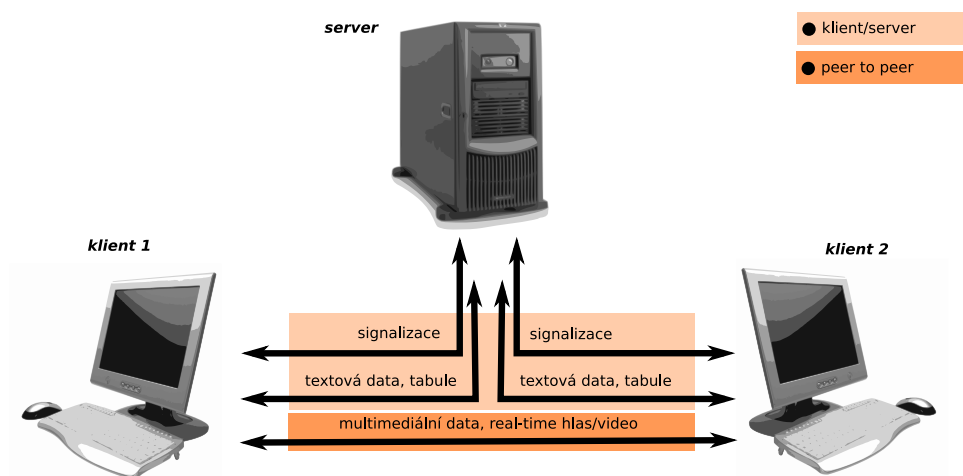
4.1 Požadované vlastnosti

Vzhledem k různým druhům dat přenášeným na sdílené tabuli je nutné komunikaci rozdělit do několika podkategorií – textová Instant Messaging část má jiné požadavky pro komunikační kanál než například přenos hlasu.

Komunikační protokol aplikace by měl proto splňovat následující vlastnosti. Jednoduchý a otevřený protokol zjednodušuje návrh síťové vrstvy aplikace a vzhledem k několika druhům přenosu je vyžadovanou vlastností flexibilita, která dále zjednoduší tvorbu aplikace. Zároveň by neměl omezovat komunikační schopnosti aplikace. Další požadovanou vlastností je odezva v reálném čase, která zvyšuje pocit interaktivity při neosobní komunikaci přes síťové médium. Vzhledem k určení postačuje soft real-time odezva – nezvládnutí operace v určeném okamžiku nemusí znamenat vážný problém, pouze snižuje požitek z komunikace.

4.2 Síťová architektura

Důležitým prvkem pro síťovou aplikaci je volba vhodné síťové architektury. Pro potřeby aplikace se jeví jako nejvhodnější kombinace dvou nejčastěji používaných architektur, tedy klient/server a peer to peer (někdy označované jako P2P, do češtiny asi nejlíp přeložitelné jako „každý s každým“). Protokol je dále rozlišován na signalizační kanál, kanál textových dat a dat sdílené tabule a kanál pro přenos real-time dat. Typické použití je znázorněno na obr. 4.1.



Obrázek 4.1: Návrh architektury

Pro signalizační část protokolu je nejvhodnější síťová architektura klient/server, která zjednodušuje navázání spojení mezi účastníky síťového sezení. Server je jednoznačně daný a v ideálním případě vždy dostupný k připojení z vnějšího světa. Slouží tedy jako prostředník pro další přenos dat i ke zřízení spojení pro peer to peer komunikaci. Dalším přenosem dat se rozumí přenos textových zpráv, komunikace pomocí tabule a jednotlivých komponent na tabuli. Peer to peer využití je především pro přenos objemnějších dat – souborů, multimediálních dat, bez nutnosti zatěžovat hlavní server. Do této kategorie patří i přenos real time hlasu a videa. Protokol by pro komunikaci měl využívat TCP/IP architektury tak, aby byl kompatibilní s použitím v síti Internet.

4.3 Protokol

Podle výše popsaných požadavků je potřeba zvolit vhodný komunikační protokol. K dispozici jsou dvě základní cesty – navrhnout vlastní komunikační protokol odpovídající požadavkům nebo zvolit již existující protokol, který vhodným způsobem tyto požadavky implementuje ve svém návrhu. Pro vlastní protokol hovoří hlavně neexistence omezení plynoucích z již existující specifikace převzatého protokolu, možnost návrhu přímo na míru danému použití a snadnější budoucí změny protokolu. Proti použití naopak hovoří náročnější návrh – protokol je potřeba řádně promyslet, protože mizivá kompatibilita s existujícími řešeními a stížená implementace v jiných aplikacích velmi snižuje využitelnost protokolu. Je možné taky tyto dvě cesty do jisté míry spojit a využít existujícího protokolu jako základu a nad ním navrh-

nout rozšíření pro potřeby aplikace. Zůstane tedy zachována hlavní vlastnost převzatého protokolu – tedy zvýšená kompatibilita s okolním světem – i možnost přizpůsobení přímo na míru aplikace.

Mezi již existujícími protokoly vhodné pro využití v aplikaci patří protokol XMPP popsaný ve třetí kapitole. Jedná se o standardizovaný a otevřený protokol, který nabízí snadnou možnost rozšíření a pro některé části aplikace je možné využít již existujících rozšíření jako například pro přenos souborů a real-time hlasu a obrazu (rozšíření Jingle a rozšíření na Jingle navazující). Další výhodou XMPP je možnost využít již existující infrastrukturu sítě Jabber – komunikace může běžet přes již zaběhnuté XMPP servery, kterých existuje nepřeberné množství, je možné využít vlastního (ve smyslu nasazení) běžícího serveru, případně spustit vlastní server a k tomu využít velké množství jak komerčních, tak open-source XMPP serverů (jako jsou Jabberd, EJabberd, OpenFire). Tato vlastnost může ušetřit nemalé finanční prostředky, které by byly zapotřebí při nasazení proprietárního serverového řešení. Jedná se jak o vývoj potřebného softwarového vybavení, tak i o čas potřebný k jeho nasazení a školení administrátorů. Je dokonce možné zprovoznit XMPP server na vlastní doméně přes zprostředkovatele bez nutnosti nasazení vlastního serveru. Tuto alternativu nabízí například společnost Google. Protokol XMPP využívá velké množství aplikací různých zaměření, od prosté formy Instant Messengeru přes aplikace podobného zaměření jako je prostředí pro spolupráci při grafickém návrhu ve vektorovém grafickém editoru Inkscape [14].

4.3.1 Přenos textových zpráv

Pro přenos běžných textových zpráv je vhodné využít běžných prostředků protokolu XMPP. Tedy použití paketu `<message>` s definovaným typem jako `normal`, `chat` případně typem `groupchat`. Jednou z výhod tohoto řešení je umožnění sledování alespoň části komunikace uživateli využívajícím běžné klienty, případně neimplementujícím všechny dále popsané možnosti komunikace.

4.3.2 Kreslení na tabuli

Kreslení na tabuli (tzv. whiteboarding) není v současné době součástí žádného oficiálního rozšíření protokolu XMPP. Každá aplikace využívá svého vlastního protokolu. Existují návrhy rozšíření a obecně panuje shoda v několika oblastech. Jedná se především o využití formátu SVG (Scalable Vector Graphics), o který se stará organizace W3C. SVG je otevřený standard pro vektorovou grafiku především pro použití v oblasti Internetu a jako XMPP je založený na technologii XML. Grafický vektorový editor Inkscape implementuje svůj vlastní protokol Inkboard [14], využívající XMPP a SVG. Další existující implementace je v Instant Messengeru Psi a implementuje protokol SVGWB [4], který je popsán v dokumentu SVG Whiteboarding od Joonase Govenia. Jedná se o jeden z návrhů oficiálního rozšíření protokolu XMPP a tímto protokolem se bude tento text podrobněji zabývat v rámci kapitoly implementace aplikace.

SVGWB používá pro rozšíření standardního protokolu XMPP zapouzdření v paketu `<message>` typu `chat`. Samotný paket kreslení na tabuli je element `<wb>` v jmenném prostoru definovaném jako `http://jabber.org/protocol/svgwb` a typu `chat`.

Protokol by měl řešit následující úkoly: pozvánky k sezení, přijetí či odmítnutí pozvánky, připojení k existujícímu sezení, odeslání aktuálního stavu dokumentu a odeslání a přijímání SVG elementů dokumentu.

Využívá definice formátu SVG 1.1 [12] v podmnožině Tiny, které musí klientský software umět zpracovat.

4.3.3 Real time zvuková/obrazová komunikace

Pro hlasovou a obrazovou komunikaci v reálném čase je využitelné především rozšíření Jingle a rozšíření definující přenos hlasu a videa pomocí RTP (Real-time Transport Protocol) přes síťové sezení navázané v rámci Jingle. XMPP protokol je použit jako signalizační protokol.

4.3.4 Ostatní multimediální data

Ostatní uživatelská multimediální data jsou především interaktivní objekty, které je možné umístit na tabuli.

Objekt je potřeba do dokumentu vložit standardním způsobem a zajistit možnost manipulace s vybraným objektem. Zároveň by měl mít objekt vlastní autonomitu v možnosti komunikace v rámci instancí objektů v síťovém prostředí. Tento problém se tedy dělí na dvě části:

- samotnou manipulaci s objektem, jako je jeho vkládání, mazání a přesun a
- přenos vlastního obsahu objektu.

V rámci SVG dokumentu je možné využít elementu `<foreignObject>`, který tak jako protokol XMPP dovoluje vkládat další XML poddokumenty ve vlastním jmenném prostoru.

Vlastní obsah objektu může opět rozšiřovat protokol XMPP ve vlastním jmenném prostoru. Pro standardní objekty, které nemají velké nároky na komunikaci, je vhodné definovat jednoduché rozšíření na bázi změny stavů objektu. Každý objekt je přesně identifikován a díky tomu je možné zprávy směřovat na správné místo a není potřeba řešit vlastní protokol. Objekty samozřejmě pro svůj obsah mohou využívat vlastní protokol zcela nezávislý na XMPP. Zamýšlené interaktivní objekty mohou být jednoduché hry, přehrávač videí z YouTube, který se vloží spolu s adresou videa nebo například textový dokument či tabulka.

4.4 Knihovny pro real-time komunikaci

Pro real-time komunikaci pomocí protokolu XMPP existuje několik rozličných knihoven a frameworků. Není tedy třeba implementovat celý protokol od začátku a komunikační část aplikace je možné stavět na ověřeném základu.

4.4.1 Iris XMPP

Knihovna Iris XMPP vznikla jako součást projektu Delta XMPP, který si klade za cíl tvorbu IM XMPP aplikací. Součástí projektu je jak samotná knihovna, tak i Ambrosia XMPP server a především populární klient sítě Jabber Psi.

Iris podporuje všechny požadavky a vlastnosti jádra protokolu XMPP definovaného v RFC 3920 – Extensible Messaging and Presence Protocol (XMPP): Core včetně zabezpečení přenosu pomocí kryptografického protokolu TLS (Transport Layer Security) a jeho předchůdce SSL (Secure Sockets Layer). Důležitou součástí pro ověření identity uživatelů je přihlašování s autentifikací pomocí SASL (Simple Authentication and Security Layer). Navíc je k dispozici i tunelování přes protokoly HTTP a SOCKS5. Toho se využívá, pokud je z nějakého důvodu zakázán průchod XMPP paketů v síti.

V rámci IM části protokolu definovaného v RFC 3921 – Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence podporuje výměnu zpráv, zasílání a příjem zpráv o prezenci, správu seznamu kontaktů (tzv. rosteru v terminologii sítí Jabber) a přihlašování k odběru prezence. Z rozšíření využitých v projektu podporuje hlavně přenos souborů a whiteboarding – tedy protokol využívaný pro kreslení na sdílenou tabuli.

Mezi hlavní výhody této knihovny patří to, že je postavená na architektuře Qt/C++ a tedy velmi dobře zapadá do konceptu Qt/C++ aplikace, jakou tento projekt je. Využívá všech hlavních vlastností této architektury, jako jsou standardní Qt signály a QCA (což je Qt Cryptographic Architecture) – knihovna pro kryptografii v Qt vytvořená taky v rámci projektu Delta XMPP.

4.4.2 Telepathy

Telepathy je kompletní jednotná specifikace pro rozšíření různých aplikací o real-time komunikaci. Je definovaný v rámci projektu freedesktop.org. Telepathy přináší následující výhody oproti běžným knihovnám:

- Real-time. Telepathy nabízí real-time komunikaci pomocí instantních zpráv a hlasové a video komunikace.
- Unifikace. Různé aplikace mohou využívat výhod Telepathy pro jejich spolupráci.
- Framework. Projekt Telepathy je složen z mnoha jednodušších podsystémů, které spolu spolupracují.

Pro komunikaci s Telepathy aplikace využívají rozhraní D-Bus. Existující implementace frameworku/specifikace jsou Tapioca a Decibel. Decibel vzniká v rámci projektu KDE 4, bohužel se stále nachází ve stavu vývoje a není jednoduše použitelný do běžných KDE aplikací. Pro budoucí využití by mohla implementace Decibelu do projektu přinést lepší propojení s prostředím KDE.

Kapitola 5

Návrh uživatelského rozhraní aplikace

Základním prvkem každé aplikace, která interaktivně komunikuje s uživatelem, je vhodné uživatelské rozhraní. V prvních počítačích se úkoly zadávaly dávkově například pomocí děrných štítků a na jeho výstup se muselo čekat podle typu úlohy i několik dnů. Nedalo se tedy hovořit o interaktivní práci a práce celá se nepodobala dnešnímu využití počítačů. Zlom nastal v době rozšiřování prvních terminálů, které byly vybaveny vstupním zařízením (například klávesnicí) a výstupní zařízením (monitor, tiskárna). Tyto zařízení dovolovaly vstup dat interaktivně bez nutnosti využití nepraktických děrných štítků, ač v počátcích byly úlohy stále zpracovávány dávkově. Skutečná potřeba uživatelských rozhraní nastala s příchodem operačních systémů, které umožňovaly sdílení prostředků počítače v reálném čase a práci více uživatelů v jeden okamžik. Jednalo se o revoluci v ovládání počítačů a využívalo se především textové komunikace. Časem ovšem došlo k dalšímu vývoji a objevily se systémy pracující v grafickém režimu a dnes již si pro běžné ovládání počítače nedovedeme představit práci bez oken a myši (samozřejmě textová konzole může být v mnohých prostředích – např. administrace serverů – stále velmi mocný nástroj). Grafické rozhraní ale zvyšuje nároky na správný návrh.

Při návrhu uživatelského rozhraní je potřeba se zaměřit především na tyto vlastnosti:

- uživatelská přívětivost – aplikace musí uživateli nabídnout jednoduché a snadno pochopitelné uživatelské rozhraní, ale nesmí to být na úkor omezení funkčnosti,
- snadná použitelnost – vychází z prvního bodu a je vázána na uživatelské rozhraní a návrh aplikace,
- multiplatformnost – aplikace by neměla být závislá na jednom konkrétním operačním systému případně grafickém prostředí.

5.1 Představení aplikace

5.1.1 Název aplikace

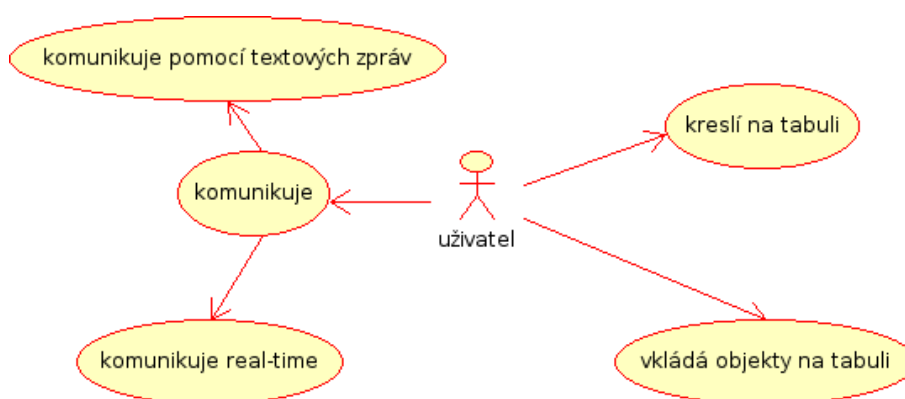
Název aplikace – Makneto – je odvozen od vzoru ve skutečném světě, tedy magnetické bílé tabule. Stejně tak jako na skutečnou tabuli je možné přidávat magnety např. pro přichycení vzkazu, tak i ve virtuální sdílené tabuli je možné na její plochu přidávat maknety.

Záměna písmena g za písmeno k je pouze symbolická a vychází ze zvyku označování aplikací v prostředí KDE.

5.1.2 Využití aplikace

Využití navržené aplikace je především pro komunikaci v rámci sítě, ale samozřejmě by nemělo být omezené jen na tuto oblast. Aplikace by tudíž měla být využitelná i bez připojení například k interaktivním tabulím.

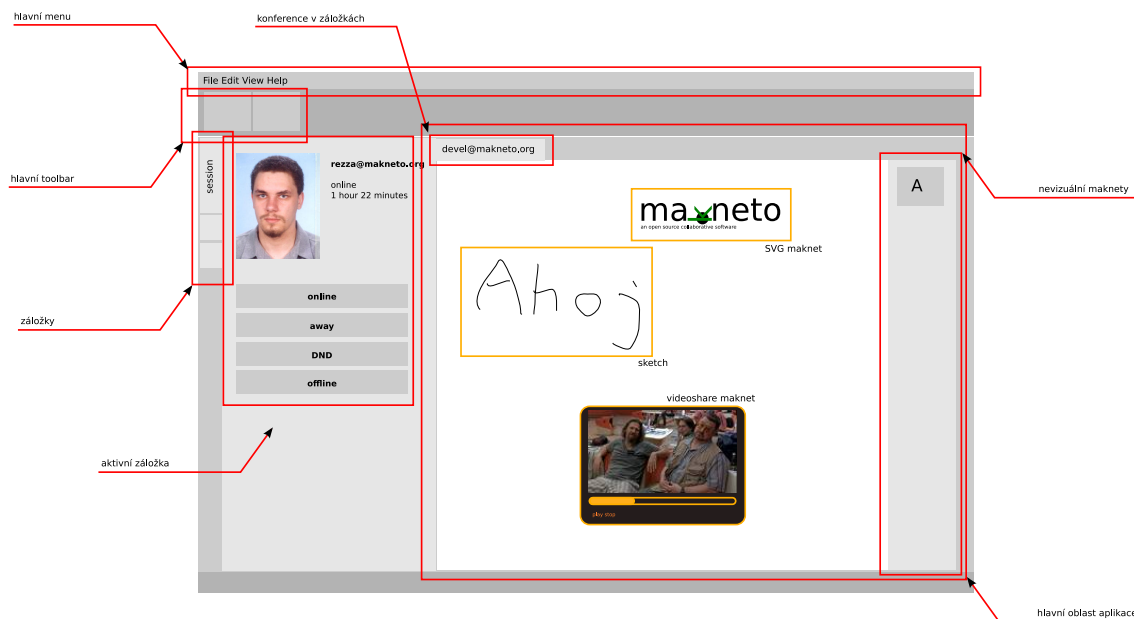
Aplikace by měla uživateli nabízet komunikaci pomocí textových zpráv, kreslení na tabuli, vkládání „maknetů“ na tabuli a real time komunikaci – hlasovou, video atd., viz obr. 5.1. Zaměření aplikace se pak dá shrnout do následujících hlavních bodů: jako komunikační prostředek k utřídění a organizaci myšlenek mezi více uživateli, jako prostředek k výuce studentů (kteří mohou interaktivně vstupovat do výuky) a k tvorbě prezentací.



Obrázek 5.1: Diagram případů použití uživatelského rozhraní

5.2 Návrh uživatelského prostředí aplikace

Uživatelské rozhraní je navrženo jako jedno velké hlavní aplikační okno s tím, že ovládání je řešené pomocí panelu se záložkami a zbývající plocha je vyplněna sdílenou plochou tabule. Jedná se tedy o odlišný přístup oproti běžným IM aplikacím, které mívají samostatné okno se seznamem kontaktů a komunikace probíhá v jiném okně. Pro textové zprávy formou rozhovoru se jedná o vhodný koncept, ale pro aplikaci kombinující několik komunikačních metod s hlavním zaměřením na sdílení plochy o řešení nevhodné. Problémem rozdělení do více oken by byla nepřehlednost. Hlavní oblastí aplikace je tedy kreslicí plocha tabule (v IM seznam kontaktů), na kterou je možné převést veškerou komunikaci. Multimediální obsah je na ní reprezentován systémem magnetů na tabuli. Uživatel tedy neztrácí přehled a ke všemu má přístup na jednom místě. Seznam kontaktů je v tomto rozvržení dostupný v rámci jedné ze záložek postranního panelu. Celé rozvržení tak jak je navrhované je vidět na obr. 5.2.



Obrázek 5.2: Návrh hlavního okna aplikace

5.2.1 Popis návrhu uživatelského rozhraní

Hlavní okno aplikace

Hlavní aplikační okno aplikace je navrženo jako kombinace rozhraní SDI (Single Document Interface) a MDI (Multiple Document Interface). Rozhraní SDI má výhodu především v přehlednosti – aplikace neotvírá několik nezávislých oken, které matou uživatele. Přesto je vhodné zavést prvky rozhraní MDI a to pomocí záložek (někde označovaných jako panely). V případě více otevřených sezení sdílené tabule by opět mohla vzniknout nepřehledná situace více otevřených oken. Takto jsou všechny uživatelské sezení organizovány v jednom okně. Hlavní okno pak panelům kreslení v záložkách umožňuje přístup k připojení k síti, seznamu kontaktů atd. Uživatel si může samozřejmě dále otevřít další hlavní okno a v něm opět další záložky. Hlavní okno dále obsahuje (viz obr. 5.2) aplikační nabídku, postranní panely a kreslicí plochu. Cílem tohoto návrhu je maximální využití prostoru okna aplikace.

Hlavní menu a panel nástrojů

Hlavní menu umožňuje ovládání aplikace. Rozdělení je běžně používané na několik položek hlavního menu (soubor, editace, nápověda) a položky podmenu. Panel nástrojů obsahuje nejčastěji přístupované položky hlavního menu tak, aby k nim byl rychlejší přístup. Významné položky mají přiřazenou klávesovou zkratku.

Postranní panely

Pro ovládání aplikace a získání informace o aktuálním stavu aplikace jsou připraveny postranní panely. Ty mohou být buď jak společné pro všechny záložky v kreslicím okně sdílené tabule (např. stav a ovládání připojení), tak kontextové – závislé na aktuálně prováděné činnosti a pracující s aktuální záložkou (tedy aktuální komunikací, např. zobrazení uživatelů

v daném sezení). Postranní panely se zobrazují v levé části hlavního okna aplikace a vzhledem k úspoře místa pro plochu kreslicí tabule musí být možné je na přání uživatele skrýt. Minimálně požadované panely, které je potřeba do aplikace začlenit, jsou: panel informující o stavu připojení s možností změny stavu pomocí ovládacích prvků (např. tlačítek) a panel se seznamem kontaktů. Pro ovládání aktuálního sezení a zobrazení informací o tomto sezení je vhodný kontextový panel sezení.

Kreslicí plocha sdílené tabule

Kreslicí plocha je hlavním prvkem okna aplikace. Má obsluhovat všechny činnosti týkající se komunikace pomocí sdílené tabule. Jsou to kreslení na tabuli pomocí různých nástrojů (pero, geometrické tvary) a vkládání objektů jako jsou obrázky a videa. Plocha k tomu určená by měla být co největší, takže se očekává implementace posuvníků. Součástí je i panel s prvky, které jsou nevizuální. Může se například jednat o znázornění hlasové komunikace. S objekty na ploše se manipuluje běžným způsobem jak je zvykem v podobných kreslicích aplikacích. Myš se provádí výběr objektu a jeho transformace (posun atd.) pomocí levého tlačítka a pravé tlačítko zpřístupňuje kontextové menu pro jednotlivé objekty.

5.3 Knihovny Qt a prostředí KDE 4

5.3.1 Qt 4

Qt[5] je oblíbená sada nástrojů a aplikační rozhraní (API) pro tvorbu multiplatformních aplikací, nyní ve verzi 4, od norské společnosti Trolltech. Původně se jednalo především o knihovny pro návrh grafického uživatelského rozhraní, ale aktuální verze umožňuje vývoj aplikací bez GUI s tím, že stále dává vývojářům k dispozici zajímavé a propracované nástroje. Qt je navrženo v jazyce C++, ale použití se neomezuje pouze na C++, ale pomocí tzv. bindings je možné využít např. v Pythonu (PyQt), Javě a dalších jazycích.

Použití Qt 4 je pro implementaci aplikace velmi výhodné z důvodu vynikající podpory tvorby grafického uživatelského rozhraní a podpory multimediálních technologií a grafických operací. Licenčně je nabízena pod tzv. duální licenci – pro komerční využití je k dispozici placená verze s podporou, pro využití v open source produktech je k dispozici verze pod licenci GNU GPL verze 2.

5.3.2 KDE 4

KDE je výkonné grafické desktopové prostředí pro Linux a další Unixové operační systémy. Projekt založil v roce 1996 Matthias Ettricht na základě knihovny Qt. V lednu 2008 se očekává vydání verze 4.0 založené na platformě KDE 4. Pro projekt je výhodné zapojení KDE, protože sjednocuje ovládání a vzhled aplikací a tedy zjednodušuje život uživatelů. KDE je uvolněno pod licenci GNU GPL verze 2.

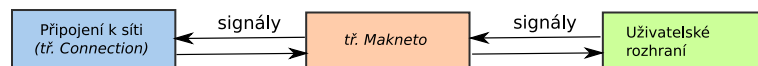
Kapitola 6

Implementace

V této kapitole je popsána implementace aplikace navržené v předchozích kapitolách. Pro implementaci je zvoleno prostředí knihoven Qt 4.3 a KDE verze 4.0. Aplikace je implementována v jazyce C++ za využití prostředků, které jsou součástí zmiňovaných knihoven. Jedná se jak o rozšíření jazyka C++ pomocí konceptu signálů a slotů (jejich použití je popsáno níže), tak i nástroje usnadňující návrh grafického uživatelského rozhraní, knihovna pro práce s multimediálním obsahem Phonon atd. Síťový XMPP klient aplikace je implementován pomocí knihovny Iris. Jsou taky zdůrazněné případné změny od návrhu a diskutovány problémy během vývoje a naznačen směr, kterým bude vývoj aplikace dále pokračovat.

6.1 Návrh aplikace

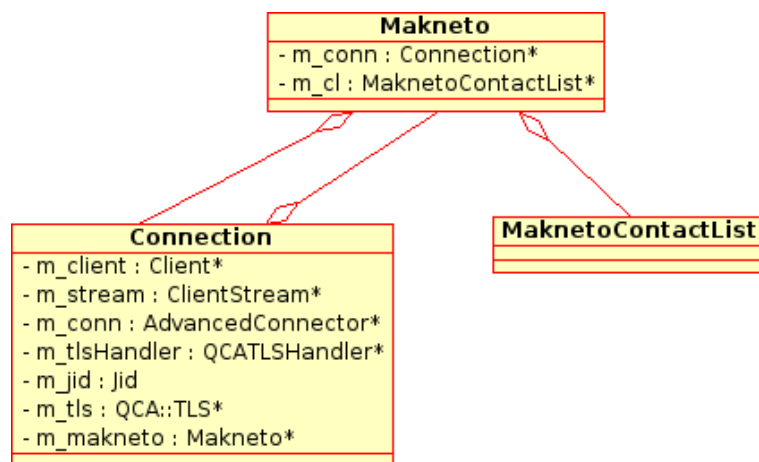
Aplikace je navržena s grafickým uživatelským rozhraním, které slouží k ovládání síťové části aplikace. Komunikace probíhá přes hlavní třídu aplikace *Makneto* (viz obr. 6.1), která zpřístupňuje grafickému rozhraní síťové rozhraní a seznam kontaktů. Ke komunikaci se vesměs využívají signály a volání metod.



Obrázek 6.1: Návrh aplikace

6.1.1 Třída Makneto

Třída *Makneto* slouží v aplikaci jako „most“ mezi uživatelským rozhraním a síťovou částí aplikace. Její třídní diagram je naznačen na obr. 6.2. Instance třídy je vytvořena jako první po startu aplikace a jako parametr předána do komponent uživatelského rozhraní (přes *MaknetoMainWindow*). Připojení (třída *Connection*) je součástí této třídy a další komponenty se mohou dotazovat na aktivní připojení pomocí metody *getConnection()*, která vrací ukazatel na instanci třídy *Connection*. Takto je možné napojit signály z uživatelského prostředí na síťovou vrstvu a opačně. Třída *Makneto* dále zpřístupňuje pro obě části aplikace seznam kontaktů (třída *MaknetoContactList*).



Obrázek 6.2: Diagram tříd Makneto

6.2 Síťová část aplikace

Základním prvkem aplikace je připojení k serveru pomocí protokolu XMPP, který je podrobně popsán ve třetí kapitole této práce. Využívá se sady tříd knihovny Iris XMPP, které jsou zakomponovány ve stromu zdrojových kódů v adresáři iris.

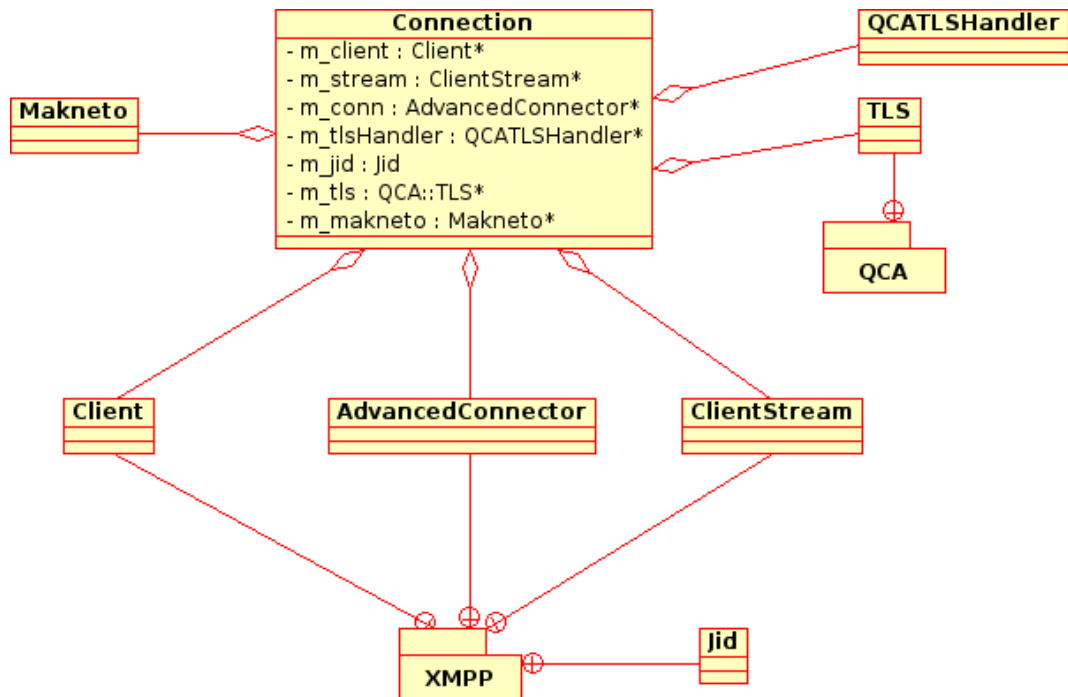
6.2.1 Třída Connection

O vlastní konektivitu se v rámci aplikace stará třída Connection, která definuje metody a signály pro komunikaci s XMPP serverem. Jedná se především o samotné připojení k serveru, nastavení prezenze, stáhnutí seznamu kontaktů (rosteru) ze serveru, přihlášení k odběru prezenze ostatních uživatelů v kontakt listu a především posílání a přijímání zpráv (textových, signalizačních a kreslicích).

Na obr. 6.3 je znázorněn diagram tříd pro třídu Connection. Třída pro svou činnost využívá třídy ClientStream, AdvancedConnector a Client z jmenného prostoru XMPP, pro šifrování využívá třídy TLS (Transport Layer Security) z balíku QCA (Qt Cryptographic Architecture). Jejich použití bude dále vysvětleno v části připojení k serveru XMPP.

Připojení k serveru

Pro připojení k serveru je v knihovně Iris připravená třída Client (v jmenném prostoru XMPP), která zapouzdřuje základní operace nad protokolem XMPP. Ve třídě Connection je jako privátní proměnná `m_client` s inicializací v jejím konstruktoru. Při inicializaci klienta se dále vyplní základní informace (název klienta atd.) a jeho vlastnosti. Zároveň se v aplikaci předpokládá přenos souborů a tedy je nutné ho povolit.



Obrázek 6.3: Diagram tříd připojení

Dále si ukážeme inicializaci klienta XMPP pomocí knihovny Iris.

```

m_client = new Client();

m_client->setClientName(„Makneto“);

QStringList features;
features << „jabber:x:data“;
m_client->setFeatures(Features(features));

m_client->setFileTransferEnabled(true);

```

Přitom je potřeba při inicializaci správně propojit signály a jim náležející sloty. Knihovna Iris využívá metody signálů a slotů knihovny Qt. Třída, která chce využívat signálů musí být potomkem třídy QObject. Toho se dosáhne pomocí makra Q_OBJECT. Ve vlastní třídě, která je používá, jsou signály i sloty obyčejné metody, pouze označené klíčovým slovem signals či slots. Propojení se dělá pomocí funkce connect, kdy je daný signál objektu napojen na slot tohoto či jiného objektu. Jedná se o rozšíření jazyka C++ pomocí makroprocesoru.

Mezi nejdůležitější patří tyto:

- `messageReceived(const Message &)` – klient přijal zprávu Message.
- `rosterItemAdded(const RosterItem &)` – přidán prvek RosterItem do rosteru. Podobně `rosterItemUpdated(const RosterItem &)` a také `rosterItemRemoved(const RosterItem &)`, jen s tím, že se jedná o aktualizaci a odstranění prvku z rosteru.

- `resourceAvailable(const Jid &, const Resource &)` – Resource pro daný JID změnil stav na dostupný.
Obdobně `resourceUnavailable(const Jid &, const Resource &)`.
- `messageReceived(const Message &)` – klient přijal zprávu Message.

Pro připojení k serveru je nejdříve nutné pro klienta vytvořit datový proud – ClientStream, kterému se nastaví vlastní Connector (v případě aplikace AdvancedConnector, který nabízí pokročilé vlastnosti, jako je využití proxy serverů apod.). Následující příklad ilustruje připojení k serveru. Nejdříve je připraven identifikátor uživatele (tedy JID podle nastavení aplikace). Datový proud bere jako svůj parametr Connector a TLS Handler, pro vytvoření šifrovaného spojení. Klient pak takto vytvořený datový proud použije k připojení k serveru.

```
m_jid = Jid(Settings::jabberID());

m_conn = new AdvancedConnector(this);
m_stream = new ClientStream(m_conn, m_tlsHandler);

Jid j = m_jid.withResource(„Makneto“);

m_client->connectToServer(m_stream, j, true);
```

V této chvíli je klient připojen k serveru a je možné se serverem komunikovat a provádět povolené operace. Bez autentifikace je možná například registrace nového uživatele. Ten musí nejdříve získat své registrační údaje. Klient proto přechází do fáze autentifikace uživatele. K ní je vyžadován identifikátor uživatele (JID) a heslo. Pokud jsou údaje správné, vytvoří se sezení (session), které spojí s daným JID a zdrojem Makneto. Uživatel je nyní připojen a autentifikován k serveru. Následně si klient vyžádá seznam kontaktů (pomocí iq dotazu, viz kapitola o protokolu XMPP) ze serveru a přihlásí se k odběru jejich prezence. Připojení k serveru je dokončeno, jakmile klient nastaví svůj vlastní stav.

Nastavení připojení

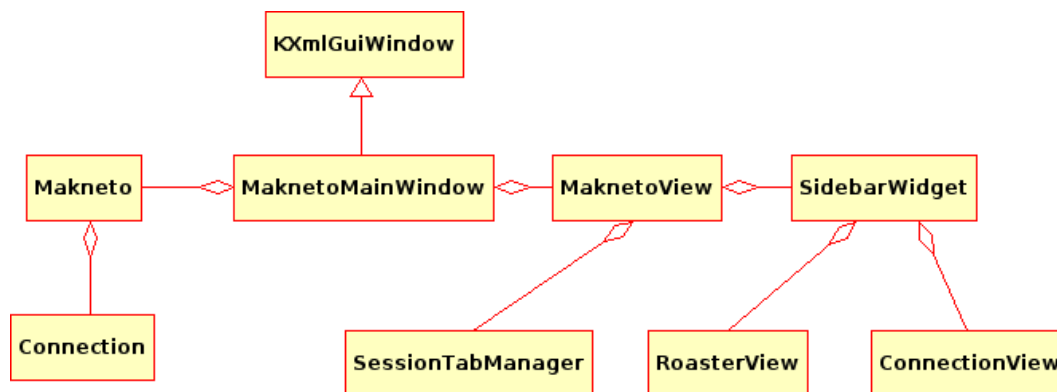
Nastavení připojení (uživatelské jméno, heslo atd.) se čerpá z konfiguračního souboru, viz. kapitola 6.7.

6.3 Uživatelské rozhraní

Uživatelské rozhraní je navrženo s cílem maximální přehlednosti a jednoduchého použití. Aplikace se skládá z hlavního okna aplikace, menu, postranních panelů a pracovní plochy. Pracovní plocha je zpracována formou záložek, jak je dnes obvyklé především ve webových prohlížečích a po spuštění je taky v záložce Home zobrazena HTML stránka projektu. Jedná se tak o kombinaci SDI (Single Document Interface) a MDI (Multiple Document Interface) rozložení aplikace. Všechna sezení jsou otevřena v rámci jednoho okna, ale rozdělena do záložek. Postranní panely se nachází na levé straně, jejich přepínání je opět realizováno pomocí záložek.

6.3.1 Hlavní okno aplikace

Hlavní okna je implementováno ve třídě `MaknetoMainWindow`. Stará se hlavně o zobrazení hlavní nabídky a zpracování požadavků z nabídky. Jako centrální objekt je nastaven objekt třídy `MaknetoView`. V něm je vytvořeno rozhraní aplikace přes grafické prvky `SidebarWidget` (správce postranních panelů) a `SessionTabView` (správce kreslicích ploch). Návrh uživatelského rozhraní včetně napojení na síťové rozhraní je na obr. 6.4.



Obrázek 6.4: Diagram tříd rozhraní aplikace

6.3.2 Postranní panely

Postranní panely jsou spravovány pomocí záložek. O jejich správu se stará grafický widget `SidebarWidget`. Pro záložky používá widgetu `KMultiTabBar` a pro ukládání a zobrazování panelů `QStackedWidget`. Jedná se tedy vlastně o zásobník postranních panelů, které jsou podle vybrané záložky vysouvány do popředí. Panely je možné úplně schovat a viditelné jsou pak pouze ikony panelů.

6.3.3 Panel připojení

Panel připojení (`Connection`) je, jak už název napovídá, určen pro ovládání připojení k serveru a nastavování stavu připojení a informacemi o aktuálním připojení. Nastavení stavu je realizováno sadou tlačítek.

Případy použití

- Uživatel chce znát aktuální stav připojení.
- Uživatel se chce připojit k serveru.
- Uživatel chce změnit stav připojení.

6.3.4 Panel sezení

Panel sezení (`Session`) obsahuje informace o aktuální vybrané relaci. Aktuálně není implementován.

6.3.5 Panel se seznamem kontaktů

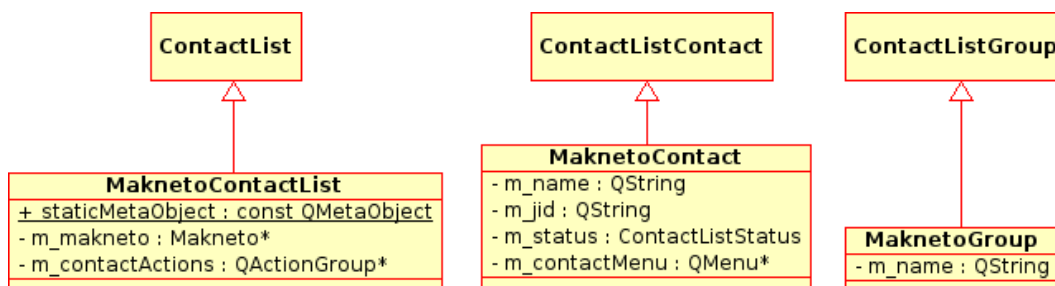
Panel se seznamem kontaktů obsahuje kontakty uživatele (tzv. roster) a je uchováván na serveru. Při připojení je přenesen a zobrazen uživateli. V aplikaci se neuchovává a proto není dostupný v režimu bez připojení k serveru. Na rozdíl od běžných IM klientů není základním kamenem uživatelského prostředí.

Případy použití

- Uživatel chce mít seznam kontaktů.
- Uživatel chce znát aktuální stav kontaktů.
- Uživatel chce přidat nového uživatele do seznamu kontaktů.
- Uživatel chce vyhledávat uživatele podle jména.
- Uživatel chce filtrovat uživatele podle stavu.
- Uživatel chce začít nové sezení s vybraným uživatelem.
- Uživatel chce znát informace o vybraném uživateli.

Třída MaknetoContactList

O správu kontaktů se stará třída MaknetoContactList. Využívá se architektury Model/View – kontakt list má svůj model (ContactListModel), který spravuje data kontaktů a pohled (ContactListView), jenž se stará o zobrazení dat. Jednotlivé kontakty (MaknetoContact) jsou tříděny do skupin (MaknetoGroup). Využívají dědění z obecných tříd ContactList, jak je znázorněno na diagramu tříd na obr. 6.5. Samotný kontakt list v postranním panelu je zapouzdřen jako widget RosterView. Tento prvek dovoluje uživateli přidávat nové kontakty, filtrovat odpojené uživatele, vyhledávat je v seznamu a další činnosti naznačené v případech použití seznamu kontaktů.



Obrázek 6.5: Diagram tříd seznamu kontaktů

6.3.6 Zobrazení přes celou obrazovku

Důležitým prvkem použitelnosti aplikace je i maximální plocha pro samotné kreslení, které je základní částí aplikace. Proto se v návrhu uživatelského rozhraní maximálně dbá na možnost rozšíření kreslicí plochy. Toho je dosaženo pomocí přepnutí do celoobrazovkového

módu (klávesovou zkratkou Ctrl+F), schováním hlavního menu aplikace (Ctrl+M) a postranních panelů (pokliknutím na ikonku právě vybraného postranního panelu). Dále je možné schovat v menu statusbar. Uživatel tak získá prostor pro vlastní komunikaci s tím, že je jednoduše možné vrátit se do výchozího stavu aplikace, případně zobrazit pouze požadovanou komponentu, jako je postranní panel pro změnu stavu připojení.

6.3.7 Grafika

Grafické prvky, především ikony, které nejsou k dispozici ve standardní distribuci prostředí KDE jsou vytvořeny v programu Inkscape a uloženy ve vektorovém formátu SVG. Následně je proveden export do rastrového formátu PNG ve všech velikostech používaných v prostředí KDE. O správnou instalaci se stará makro kde4_install_icons. Ikona aplikace je opět vytvořena v programu Inkscape.

6.4 Kreslení na tabuli

Kreslení na tabuli se skládá ze dvou částí – síťové komunikace a uživatelského rozhraní. Obě části pak musí být vzájemně vhodně propojeny.

V projektu je využito implementace whiteboardingu z IM klienta Psi od Joonase Goveniuse s úpravami nutnými pro začlenění do aplikace a s rozšířením o požadovanou funkčnost. Jedná se především o vlastní správu připojení a komunikaci s aplikací a síťovou částí, což je nejdůležitější komponenta sdílené tabule. Dále jsou to pak úpravy pro prostředí KDE.

6.4.1 Síťová komunikace

Komunikace probíhá následujícím způsobem. Síťový klient přijme zprávu, která obsahuje element protokolu SVGWB [4] a vyšle signál `connMessageReceived` se zprávou `Message`. Zpráva obsahuje XML element `wb`, ke kterému lze přistupovat a ten se dále předá objektům kreslicí plochy, v tomto případě `WbScene` metodou `processWb`. Opačným směrem je element kreslení zabalen do zprávy `Message` a standardním způsobem předán síťovému klientovi reprezentovanému třídou `Connection`.

Příkazy protokolu jsou:

- `New` – vytvoří nový objekt.
- `Configure` – nastavení parametrů objektu.
- `Move` – přesun objektu.
- `Remove` – smazání objektu.

6.4.2 Příkaz „new“

Příkaz „new“ je používán ve zprávě o vytvoření nového objektu na sdílené tabuli a je kódován jako XML element `<new>`. Přenáší se v rámci paketu `<wb>` a musí obsahovat právě jeden kreslicí objekt. Tento objekt je pak označen indexem (pořadím vložení na plochu sdílené tabule) a identifikátorem uživatele. Tím je zaručena unikátnost v rámci síťového distribuovaného prostředí. Jednotlivé přenášené kreslicí prvky a jejich parametry jsou uvedeny níže a odpovídají definicím základních tvarů ze specifikace SVG Tiny [12].

Prvek „path“

Prvek „path“ slouží k přenosu dat nakreslených pomocí tzv. nástroje „freehand“. Jedná se vlastně o záznam pohybu kreslicího nástroje (pera) pomocí křivky. Parametry elementu `<path>` jsou barva pera (*stroke*) a tloušťka čáry (*stroke-width*).

Geometrické prvky

Geometrické prvky jsou přímka (`<line>`), obdélník (`<rectangle>`) a elipsa (`<ellipse>`). Jako křivka všechny definují barvu a tloušťku okraje a přidávají i barvu výplně objektu. Podle tvaru jsou součástí parametrů souřadnice, které definují polohu a rozměry objektu na kreslicí ploše.

Prvek „image“

Vkládání rastrových obrázků (SVG element `<image>`) je operace náročnější, protože je potřeba přenést i jeho data. Element obrázku má atribut *href* (z jmenného prostoru *xlink*) a v něm jsou přenesena data zakódovaná pomocí base64 kódování. Díky tomuto kódování je možné je vložit přímo do SVG dokumentu.

6.4.3 Příkaz „configure“

Příkaz „configure“ se používá k nastavení vlastností objektu. Ten je specifikován v parametru *target* XML elementu `<configure>`. Samotné atributy se mění pomocí elementu `<attribute>`. V parametru *name* je specifikován název měněného atributu a novou hodnotu obsahuje tělo elementu.

6.4.4 Příklad komunikace

V následující ukázce protokolu je vytvořena kružnice na pozici [100, 100] s poloměrem 20 a tloušťkou čáry 1. Kružnici pak jsou změněny vlastnosti tak, že její tloušťka je tři pomocí příkazu `configure`. Mění se její atribut `stroke-width`. Následně je objekt vymazán.

```
<wb xmlns=„http://jabber.org/protocol/svgwb“ session=„s1“ >
<new id=„1/rezza@jabber.cz“ index=„1“ >
<circle cx=„100“ cy=„100“ stroke=„rgb(0,0,0)“ r=„20“
stroke-width=„1“ />
</new>
</wb>
```

```
<wb xmlns=„http://jabber.org/protocol/svgwb“ session=„s1“ >
<configure target=„1/rezza@jabber.cz“ version=„1“ >
<attribute name=„stroke-width“ >3</attribute>
</configure>
</wb>
```

```
<wb xmlns=„http://jabber.org/protocol/svgwb“ session=„s1“ >
<remove target=„1/rezza@jabber.cz“ />
</wb>
```

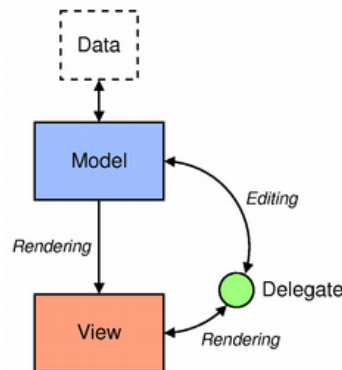
Jak je možné vidět, jednotlivé objekty mají svoje unikátní značení, přes které jsou identifikovány a dále i sezení má vlastní identifikátor. Ten se generuje jako náhodný identifikátor (v ukázkovém příkladu nastaven na `s1`). Je zapotřebí zajistit, aby se dva identifikátory v distribuovaném prostředí neshodovaly. Jako dostatečné se jeví použití času založení sezení ve spojení s Jabber identifikátorem (JID) uživatele, který ho založil. Jednotlivé objekty jsou označeny identifikátorem skládajícím se z pořadového čísla umístění na sdílenou tabuli a opět JID uživatele, který je vytvořil.

6.4.5 Uživatelské rozhraní

Uživatelské rozhraní kreslení na tabuli využívá plně všech výhod, které přináší toolkit Qt od verze 4.2, především právě v oblasti manipulace s grafikou a kreslením. Skládá se z ovládacích prvků pro kreslení a manipulaci s objekty na tabuli a samotné kreslicí plochy. Ta je implementována pomocí „The Graphics View Frameworku“.

The Graphics View Framework

The Graphics View Framework je pokročilé prostředí pro práci s grafickými objekty v aplikaci, který je založen na Model/View/Controller architektuře (viz 6.6). Ta odděluje vizuální podobu od dat, podle kterých se následně kreslí. V Qt je rozdělení na třídy `QGraphicsScene`, což je model scény a `QGraphicsView`. `QGraphicsView` je plocha, na kterou se následně kreslí objekty. Každý objekt je reprezentován jako `QGraphicsItem`.



Obrázek 6.6: Model/View architektura

Graphics View využívá BSP (Binary Space Partitioning) stromů pro ukládání scény, díky čemuž dosahuje velmi dobré rychlosti zpracování a umožňuje real-time zpracování rozsáhlých scén.

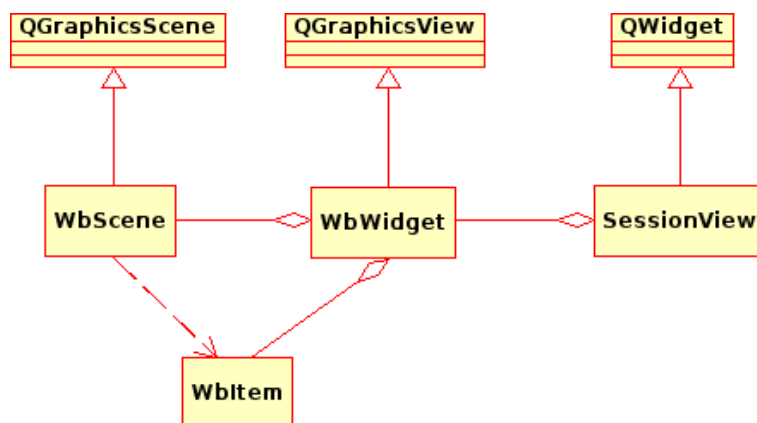
Scéna reprezentovaná skrze třídu `QGraphicsScene` je zodpovědná za rychlou správu grafických objektů, propagování událostí jednotlivým objektům a správu stavu objektů, jako je výběr a zaměření (focus) objektu.

Pohled reprezentovaný skrze třídu `QGraphicsView` přináší grafický prvek (widget) pro zobrazení obsahu scény. Pro jednu scénu může být definováno více pohledů.

Prvek (`QGraphicsItem`) je základní třída pro objekty v Graphics View Frameworku.

6.4.6 Kreslicí část okna sezení

Kreslicí část okna je součástí třídy `SessionView` a je rozdělena na panel nástrojů a samotný grafický widget pohledu. Ten je v aplikaci realizován třídou `WbWidget`, která je potomkem třídy `QGraphicsView` a má nastavený vlastní model scény. Model scény je implementován ve třídě `WbScene`, která je potomkem třídy `QGraphicsScene`. Příchozí data protokolu SV-GWB jsou předávány do scény, která je zpracovává a následně jsou zobrazeny přes pohled. Diagram tříd je znázorněn na obr. 6.7.



Obrázek 6.7: Diagram tříd kreslení na tabuli

Případy použití

- Uživatel chce kreslit pomocí předdefinovaných nástrojů.
- Uživatel chce manipulovat s nakreslenými objekty.
- Uživatel chce měnit vlastnosti nakresleného objektu – barvy, tloušťku čáry.
- Uživatel chce vybírat objekty, včetně více objektů naráz.
- Uživatel chce mazat nakreslené objekty.

Panel nástrojů

Nástroje pro manipulaci s kreslenými objekty, jako je výběr, posun, rotace, mazání a výběr kreslicích nástrojů, se nachází v horním panelu jako tlačítka s ikonami, symbolizujícími danou operaci.

6.5 Přehrávač multimediálních dat

Důležitou součástí projektu se zaměřením na sdílení multimediálních dat je jejich zobrazení cílovému uživateli aplikace. Uživatel by měl získat nad těmito daty plnou kontrolu – tedy jak nad zobrazením, tak i nad jejich ovládním, což zahrnuje operace jako jsou spuštění a pozastavení přehrávání, posun v přehrávaných datech či hlasitost u zvukových dat. Multimediálními daty jsou pro přehrávač chápány video a zvukové soubory libovolného podporovaného formátu. Přehrávač musí umět rozlišit mezi typem souboru a nabídnout buď

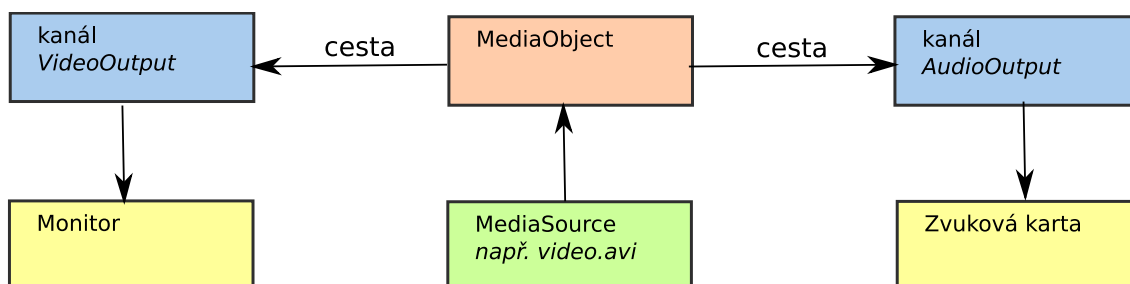
uživatelské rozhraní pro přehrávání videa s obrazem nebo pouze ovládací prvky pro zvukové soubory. Dále rozebereme jak je tato funkcionality implementována v aplikaci.

6.5.1 Phonon

Při řešení této práce se o video a zvuk stará nově uvedené prostředí knihoven Phonon [11], které usnadňuje vývoj multimediálních aplikací a přináší nové možnosti zpracování multimediálních dat. V následujícím textu je znázorněno jeho využití.

Architektura

Phonon používá koncept skládající se ze tří základních komponent – mediálních objektů (media objects), kanálů (sinks) a datových cest (paths). Mediální objekt se stará o načítání dat ze zdrojů (media sources) a o základní řízení přehrávání jako je např. jeho spuštění a pozastavení. Kanály se používají jako výstupní zařízení. Tím může být například obrazovka pro video data a zvuková karta pro audio data. Pomocí datových cest se skládá přehrávací řetězec do grafu, jímž v rámci Phononu putují data od zdroje až po výstup, případně přes další součásti systému jako jsou komponenty efektů. Každý objekt v Phononu má vstupy a výstupy a ty je možné propojovat právě pomocí cest. Video přehrávač se tedy skládá ze vstupního datového zdroje, mediálního objektu, cest a dvou kanálů – pro video a pro zvuk. Výsledný graf implementuje přehrávání. Přehrávání je řízeno mediálním objektem, který rozesílá datový proud kanálům, propojeným pomocí datových cest. Všechny uzly grafu jsou synchronizovány. Příklad sestaveného grafu pro přehrávání video souboru se zvukovou stopou je na obr. 6.8.



Obrázek 6.8: Graf video přehrávače

Samotný framework Phonon je rozdělen do dvou částí – frontend a backend. Frontendem jsou myšleny právě výše uvedené vysokoúrovňové objekty pro přehrávání médií. Backend naopak přináší nezávislost nad použitým multimediálním frameworkem a hardwarem. Samotný Phonon tedy neobsahuje žádné funkce pro práci s multimédií a využívá funkcí jiných knihoven. Ve Windows se jedná o DirectShow, v Linuxu GStreamer a Xine.

Mediální objekty a kanály

Mediální objekty (media objects) jsou základním stavebním kamenem v Phononu. Programátorovi nabízí základní obsluhu přehrávání multimediálních dat. Tyto data jsou dodávány z mediálních zdrojů (media sources). Data mohou být čtená ze souboru nebo se může jednat o datový proud přenášený po síti. Tyto objekty jsou v Phononu implementovány ve třídě MediaObject, zdroje jako MediaSource. Doplnkem je třída MediaController, které nabízí

přístup k rozšiřujícím vlastnostem média – např. ke kapitolám na DVD. Kanály jsou oproti ostatním objektům pouze výstupní a reprezentují virtuální zařízení.

6.5.2 Přehrávač

Přehrávač v aplikaci je postaven na frameworku Phonon. Uživateli je zobrazen po příjmu multimediálního souboru, který je schopen přehrát, případně ho může uživatel sám vyvolat na vybraném multimediálním souboru. Přehrávač nabízí běžné prvky uživatelského rozhraní video/audio přehrávačů.

Případy použití

- Uživatel chce přehrávat přijaté video nebo audio.
- Uživatel chce spustit/pozastavit/ukončit přehrávání.
- Uživatel se chce pohybovat ve videu/audiu.
- Uživatel chce měnit hlasitost, vypnout zvuk.
- Uživatel chce znát základní informace o multimediálních datech.

V případě streamovaného přenosu dat se případy použití omezují prakticky pouze na přehrávání a spuštění/pozastavení/ukončení přehrávání, protože k většině streamovaného obsahu nelze přistupovat sekvenčně, případně jen v nabufferované části streamu na straně klienta.

Třída MediaPlayer

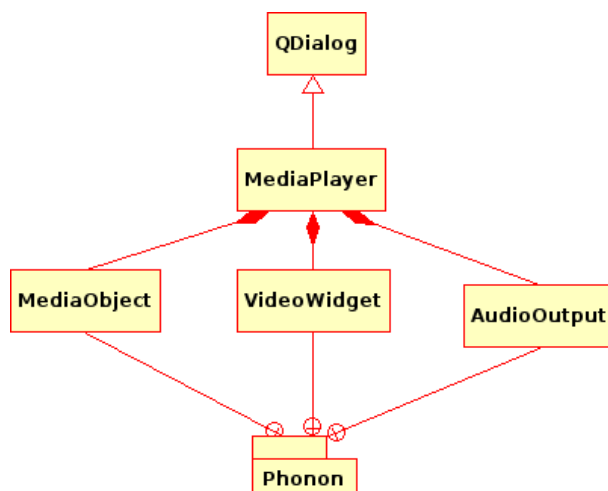
Třída MediaPlayer v aplikaci zapouzdřuje multimediální přehrávač s ovládacími prvky. Jedná se o grafický prvek odvozený z třídy QDialog, který obsahuje widget video přehrávače, ovládací prvky a informace o přehrávaném souboru. Jak je vidět na obr. 6.9, třída MediaPlayer využívá pro přehrávání třídu MediaObject. Pro zobrazení obrazových dat k tomu určený VideoWidget. Ten je možné používat stejně jako běžný QWidget (prvek uživatelského rozhraní). V případě čistých audio dat se nezobrazuje a je viditelné pouze ovládání přehrávání. AudioOutput slouží k výstupu přehrávaného zvuku. Jako výstupní zařízení se používá výchozí zařízení prostředí Phonon.

Nejdůležitější metodou je setCurrentSource, pomocí které se nastavuje aktuální zdroj pro přehrávání. Zdroj přehrávání je typu MediaSource. V knihovně Phonon jsou připravené zdroje pro přehrávání přímo ze souborů a libovolných vstupně výstupních zařízení (přes QIODevice). Další možností je vlastní datový zdroj, který je potomkem třídy AbstractMediaStream.

```
void MediaPlayer::setCurrentSource(const Phonon::MediaSource &source)
```

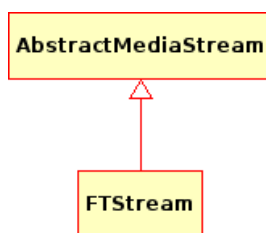
Třída FTStream

Vlastního datového zdroje se používá v aplikaci pro zobrazení tzv. streamovaných dat. Jak již bylo uvedeno výše, tato třída je potomkem třídy AbstractMediaStream, což je abstraktní rozhraní pro datové proudy pro třídu MediaObject. Tato třída musí implementovat minimálně tyto metody *reset()*, *needData()* a *enoughData()*. Pro plnění dat datového proudu



Obrázek 6.9: Diagram tříd videopřehrávače

do mediálního objektu je možné využít dvou přístupů. První je tzv. „push“, kdy jsou data neustále dodávána do mediálního objektu a ten si pouze pomocí metody enoughData() řekne, že má dostatek dat a další nejsou aktuálně potřeba. Přístup „pull“ naopak čeká na explicitní žádost o data a ty pak předává mediálnímu objektu. Třída FTStream implementuje první přístup. Jak přichází data do aplikace tak je ukládá do dočasné paměti (bufferu) a následně hned předává dál do mediálního objektu.



Obrázek 6.10: Diagram tříd datového proudu

6.6 Konfigurace aplikace

Pro konfiguraci aplikací se užívá v prostředí KDE jednotné prostředí KConfig XT [7]. Ten zjednodušuje jak správu konfigurace všech aplikací pro uživatele a administrátory – a to v prostředí KDE, které je známo svojí konfigurovatelností a tím i velkým množstvím konfiguračních možností a nastavení, tak hlavně pro programátory. Proto je využit i při implementaci popisované aplikace.

Základní prvky frameworku tvoří:

- KConfigSkeleton – třída KDE pro flexibilní přístup ke konfiguraci v aplikaci.
- Konfigurační XML soubor specifikující nastavení (kcfg).

- Řídící soubor pro překladač konfiguračního XML souboru (kcfg).
- `kconfig_compiler` – překladač, který automaticky generuje zdrojový kód v jazyce C++ z dodaných konfiguračních souborů (kcfg a kcfgc). Vygenerovaná třída je potomkem třídy `KConfigSkeleton` a nabízí API pro přístup ke konfiguraci.

V aplikaci následně stačí pouze přidat vygenerovaný hlavičkový soubor pomocí `include` a jednoduše přistupovat ke konfiguraci. Uživatelské rozhraní pro konfiguraci je vhodné vytvořit v Qt GUI Designeru – stačí jen jeho prvky pojmenovat stejně jako konfigurační parametry v kcfg souboru právě s prefixem `kcfg_`. Pro zobrazení takto vytvořeného konfiguračního dialogu se volí třída `KConfigDialog`, s uživatelským rozhráním se pracuje stejně jako s jiným vygenerovaným ui souborem.

6.7 Plánovaná rozšíření

Vzhledem k rozsahu práce se do stávající aplikace nevešlo několik funkcí, které by mohly velmi podstatně rozšířit funkcionalitu a využití celého projektu. Jedná se především o hlasovou real-time komunikaci a rozšíření sdílené tabule o další interaktivní prvky. Vývoj aplikace bude dále pokračovat v SVN repositáři (na adrese makneto.sf.net) i po dokončení diplomové práce a při návrhu aplikace byly tyto funkce brány v potaz. Oproti původnímu návrhu a implementaci se jeví jako zajímavá možnost výměny síťové části za prostředí Telepathy.

Kapitola 7

Nasazení aplikace

Součástí diplomové práce je kromě samotné aplikace a technické zprávy také vlastní správa projektu, což je velmi důležitý aspekt tvorby softwarového produktu. V navazujícím textu je popsán způsob vývoje projektu a postup jeho nasazení ke koncovému uživateli.

7.1 Open Source

Open source^[13] (označovaný jako OSS) je programové vybavení počítače s otevřeným zdrojovým kódem. Otevřenost zde znamená jak dostupnost zdrojového kódu, tak především licenci k využití tohoto kódu. Dle licence je umožněno nahlížet do zdrojových kódů, provádět jejich modifikaci (za účelem opravení chyb, přidání nových vlastností atd.) a svobodně distribuovat tyto zdrojové kódy a výslednou aplikaci.

Podmnožinou open source je tzv. svobodný software (free software), prosazovaný Free Software Foundation, který prosazuje další podmínky, například právě musí zaručovat uživatelům právo na šíření odvozeného díla původního programu.

O oficiálně uznávané licence se stará nezisková organizace Open Source Initiative. Příkladem takových licencí jsou například BSD licence, Apache licence, Mozilla licence a nejznámější GPL licence.

Vývoj projektu je veden jako open source. Každý uživatel a programátor má přístup ke zdrojovým kódům aplikace a může je libovolně upravovat či dokonce na tomto kódu založit vlastní projekt (v případě dodržení licence). Aplikace naopak těží z volně dostupných zdrojových kódů knihovny Iris, projektu Psi, knihovny Qt a prostředí KDE. Nejde ovšem jen o zdrojové kódy, ale i o celkovou otevřenost vývoje, do kterého mají možnost zasahovat i samotní koncoví uživatelé.

7.1.1 Licence GNU GPL

GNU General Public Licence (zkráceně GNU GPL nebo jenom GPL) je svobodná softwarová licence napsaná Richardem Stallmanem pro jeho operační systém GNU. GPL je tzv. copyleftová licence. Zdrojové kódy pod GPL mohou být svobodně upravovány a šířeny musí být opět pod licencí GPL. Autor musí zpřístupnit na vyžádání zdrojové kódy, obvykle jsou ovšem snadno dostupné spolu s binární distribuční verzí software.

GNU licence je k dispozici ve třech verzích, dnes je nejrozšířenější verze 2 a pomalu se začíná rozšiřovat čerstvá verze 3. Rozšířením licence GPL je licence LGPL (GNU Lesser General Public Licence), která poskytuje větší práva především v oblasti využití u knihoven.

Makneto je licencováno právě pod licencí GNU GPL verze 2 ^[3].

7.2 Vedení projektu Makneto

Cílem je pokud možno maximální zapojení komunity uživatelů. Z tohoto důvodu je projekt veden s otevřenými zdrojovými kódy tak, aby bylo pokud možno co nejjednodušší do něho přispívat. To ovšem přináší potřebu dobré organizace, přístup k souvisejícím informacím a komunikace mezi vývojáři a uživateli.

7.2.1 Správa projektu

Pro správu projektu je využito webové služby SourceForge.net. Jedná se o komplexní službu pro správu a vedení open source projektů. Nabízí prostor pro hostování zdrojových kódů, souborů, webové prezentace, wiki stránky a nástroje jako je Bugzilla, což je systém pro hlášení chyb, fórum a další. Vývojáři zde mohou spolu komunikovat a zapojovat se do různých projektů.

Projekt je k dispozici na adrese <http://makneto.sf.net/>.

7.2.2 Repositář zdrojových kódů

Projekt využívá SVN repositářů služby SourceForge.net. CVS repositář není využit.

Kompletní zdrojové kódy v aktuální verzi je možné získat pomocí Subversion následovně:

```
svn co https://makneto.svn.sourceforge.net/svnroot/makneto/trunk
```

Aktuální vývoj probíhá v adresáři *trunk*. Vydané verze v adresáři *tags*. Případné větve patří do adresáře *branches*.

7.2.3 Prezentace projektu

Prezentace projektu běží na vlastní adrese <http://www.makneto.org/> věnované komunitou včetně hostingu této prezentace a nasazení profesionálního CMS (Content Management System). Součástí tvorby prezentace bylo i grafické ztvárnění loga aplikace a vytvoření domovské stránky do integrovaného prohlížeče v Maknetu.

Cílem prezentace je přilákání nových uživatelů a komunikace se stávajícími uživateli.

7.3 Nasazení aplikace Makneto

Vývoj aplikace nekončí jen jejím překladem do strojového kódu, ale pokračuje velmi důležitou činností a to dodáním aplikace cílovému uživateli. Mnohdy se jedná o velmi náročnou činnost, která může bez problémů přesáhnout rozsah samotného vývoje. Před samotným předáním uživateli je potřeba aplikaci důkladně otestovat, provést lokalizaci, připravit uživatelský manuál a především uvést vše do takového stavu, aby mohla být jednoduše distribuována dále zákazníkovi. Jedná se většinou o neznámé prostředí, zcela odlišné od pracovního, ve kterém byla aplikace připravována. Můžou to být chybějící knihovny, neočekávané vstupy, nekompatibilní software či hardware atd. Tímto se zabývá tato kapitola, která probere hlavní úskalí nasazení aplikace ke spokojenosti koncového uživatele.

7.4 Vydání aplikace

Vydání aplikace (release) je proces, při kterém je aplikace připravována k distribuci. Samotný vydání se dá rozdělit na několik etap. První je tzv. alfa verze – první verze, určená pro testování v rámci komunity vývojářů a testerů, následuje pokročilejší betaverze, před jejímž vydáním dochází většinou k zmrazení přidávání nových vlastností (feature freeze) a opravují se pouze nalezené chyby. V následující fázi je prakticky finální verze, označovaná jako kandidát k vydání (Release Candidate). Pokud není nalezena žádná závažná chyba, stává se z Release Candidate finální verze.

V open source světě je základním heslem „Release early, release often“, tedy v překladu „vydávej brzy, vydávej často“. Tento přístup má své klady i zápory. Mezi klady patří zapojení uživatelů do vývoje software. Vzniká tak vazba, kdy i obyčejní uživatelé hlásí chyby v rané fázi vývoje a ty jsou pak rychleji opravovány. To platí i o rozšiřování nových funkcí. Mezi zápory pak patří, že se mezi uživatele šíří nedokončená aplikace, což může odradit ty uživatele, kteří nemají zájem zapojit se do vývoje.

Samotné vydání se provádí tzv. otagováním vývojového stromu trunk v SVN repositáři a přípravou balíčků zdrojových kódů pro další překlad.

Při samotném vydání je vhodné dodržet běžně používané pravidla[6].

7.5 Lokalizace a národní prostředí

Pro zvýšení šance prosazení projektu je důležité nabídnout uživatelům aplikaci i v jejich národním jazyce s plnou podporou národního prostředí. Primárním jazykem aplikace je sice angličtina (tak jako v celém projektu KDE), jinak nejpoužívanější jazyk pro komunikaci s uživatelem jak v open source světě, tak i ve sféře komerčních aplikací. V open source je hlavním průkopníkem „The Translation Project“ v rámci něhož vzniklo překladatelské prostředí GNU gettext.

Internacionalizace (v open source označovaná jako i18n) je příprava aplikace k podpoře libovolného národního prostředí než je výchozí. Obnáší to podporu zobrazení zpráv v (nejen) rodném jazyce uživatele, vstup dat od uživatele, ale i ze souborů a jiných zdrojů a národních formátů pro zobrazení datumu, času, čísel atd. Lokalizace (označována jako l10n) je pak proces adaptace internacionalizované aplikace k podpoře daného jazyka.

7.5.1 GNU gettext

Gettext je sada nástrojů vyvíjená v rámci projektu GNU. Uživatelům nabízí jednotnou konvenci jak mají aplikace přistupovat k lokalizovaným datům, organizaci lokalizovaných katalogů zpráv, knihovnu pro přístup k lokalizovaným textům a nástroje pro vytváření lokalizačních souborů.

Lokalizační soubory

Samotná lokalizace je v systému gettext řešená v souborech s překlady řetězců. Soubory s příponou .po jsou tzv. Portable Objects a jsou určeny ke zpracování člověkem v textové formě. Naopak soubory .mo jsou tzv. Machine Objects v binární formě určené pro

zpracování počítačem. Pro každý lokalizovaný jazyk existuje jeden PO soubor, který se následně překládá do MO formátu. Pro prvotní vygenerování PO souboru existuje program xgettext. Ten dokáže procházet zdrojové kódy nejčastěji používaných programovacích jazyků (především C, C++, Perl) a vyextrahovat správně označené zprávy.

Nástroje

V rámci projektu KDE vznikly velmi kvalitní nástroje pro překlad připravených PO souborů. Jmenovitě se jedná o KBabel a jeho nástupce Lokalize. Další existují i pro ostatní desktopové prostředí a některé projekty mají i online překladatelské nástroje na Internetu, což usnadňuje spolupráci více lidí na překladech.

7.5.2 Překlad aplikace Makneto

Součástí řešení projektu je vytvoření internacionalizované aplikace a lokalizace především do češtiny. Podpora národního prostředí je již přímo součástí knihoven prostředí KDE, na kterých aplikace staví. Výchozím jazykem aplikace je angličtina, jak pro zobrazitelné zprávy, tak pro manuál a veškeré informace na Internetu (domovská stránka, wiki). Všechny zprávy ve zdrojovém kódu jsou proto v angličtině. Současně je Makneto k dispozici i kompletně v češtině a základní funkcionalita i ve španělštině. Další lokalizace jsou na komunitě uživatelů, zdrojové PO soubory jsou k dispozici v repozitáři projektu. Hotové MO soubory jsou distribuovány společně s aplikací.

7.6 Distribuce

Způsob distribuce aplikace velmi záleží na jejím způsobu využití. Aplikace vyvíjená na zakázku je většinou dodaná cílovému zákazníkovi přímo na místo jejího nasazení a připravena k použití. Naopak u aplikace, jakou je Makneto která míří do širokého okruhu uživatelů, navíc v neznámém prostředí, není možné provést nasazení přímo u uživatele. Proto je potřeba využít nějakého mechanismu distribuce. Open source aplikace se distribuují formou zdrojových kódů, ale pro koncového uživatele je výhodnější ji dodat již přeloženou a snadno instalovatelnou. Situaci dále usnadňuje možnost distribuce pomocí Internetu, v současné době využívaná i pro komerční aplikace, neboť snižuje náklady na výrobu distribučního média, krabice a samotné distribuce. Pro open source aplikace se jedná vesměs o výhradní formu distribuce.

7.6.1 Specifika cílových platforem

Do způsobu šíření dále zasahuje cílová platforma, či v případě multiplatformní aplikace cílové platformy. Rozdíly mohou být velmi výrazné a velmi stěžují nasazení. Vzhledem k tomu, že primárně je cílovou platformou Makneta Linux, tak v následujícím textu bude rozebrána podrobněji právě problematika balíčků a balíčkovacích systémů.

7.6.2 Balíčkovací systémy

Jak již bylo výše popsáno – balíčkovací systémy slouží ke správě tzv. balíčků. Pomocí nich probíhá distribuce software v distribucích Linuxu. Balíčky obsahují kompletní instalaci aplikace a definici závislostí. Nejčastěji je možné narazit na balíčky formátu RPM (Red Hat Package Manager, dnes RPM Package Manager) společnosti Red Hat, který využívají i další

distribuce jako SUSE a Mandriva a balíčky Deb, které jsou využívány vesměs v systémech odvozených z GNU/Debianu. Jako je ve světě otevřeného softwaru zvykem, existuje další spousta alternativ. Mezi nejznámější správce balíčků patří systémy APT, YUM, SMART a nově Zypper v OpenSUSE. Správce balíčků řeší především závislosti balíčků, jejich stažení z Internetu (či jiného média jako CD, DVD) a aktualizace.

RPM

RPM [2] (RPM Package Manager) byl původně vyvinut firmou Red Hat pro systém Red Hat Linux, ale v současné době je využit v mnoha dalších distribucích Linuxu a dokonce portován i na další systémy jako BSD a Windows (přes Cygwin, ve verzi 5).

Princip tvorby RPM spočívá v překladu zdrojových kódů, následné archivaci přeložených souborů a vytvoření samotného RPM. K popisu tohoto procesu se používá tzv. SPEC soubor, což je vlastně „recept“, podle kterého je možné vytvořit výsledné RPM soubory (z jednoho SPEC souboru může být vytvořeno více balíčků, u knihoven se tvoří balíček se samotnou knihovnou a devel balíček pro vývojáře). Ve SPEC souboru se specifikují metainformace o balíčku, které je pak možné využít ve správci balíčků jako je název, popis balíčku, verze apod. Jedná se vlastně o obyčejný textový soubor rozdělený na sekce – sekce informací a sekce podle které se provádí příprava balíčku.

Výsledný SPEC soubor (zkrácený o výčet souborů k instalaci) pro Makneto je součástí přílohy B technické zprávy. Po zpracování vznikne instalační RPM balíček, který je možný distribuovat jak samostatně, tak zařadit do repositáře.

Build Services

Ruční správa tvorby balíčků je ovšem zdlouhavá práce, a proto pro většinu distribucí vznikají služby, které automatizují mnohé úkony spojené s tvorbou a distribucí balíčků. Jmenovitě se jedná o OpenSUSE Build Service od Novellu, Koji Build System z projektu Fedora, PPA v Ubuntu atd. Požadavkem zadavatele práce (společnost Red Hat) je distribuční balíček v repositáři Fedora Extras. Vzhledem k tomu, že OpenSUSE Build Service nabízí tu samou funkcionalitu a navíc pro více různých distribucí, tak jsou distribuční balíčky připravovány touto službou. Služba velmi usnadňuje překlad a distribuci projektu a proto je v této práci zmíněná jako velmi přínosná.

Kapitola 8

Závěr

Cílem projektu bylo navrhnout aplikaci, která umožní spolupráci více uživatelů pomocí interaktivní multimediální tabule. V současné době je tato funkčnost implementována – v aktuální verzi v době dokončení odevzdání diplomové práce je k dispozici kreslení na sdílenou tabuli, textová komunikace přes protokol XMPP a streamovaný přenos multimediálních dat. Uživatelé mohou tedy nyní sdílet a prezentovat svoje myšlenky v přirozené podobě bez nutnosti fyzické přítomnosti na jednom místě. Využití se nabízí především při vývoji dalších open source aplikací, kdy někdy může jeden náčrt na sdílené tabuli znamenat pochopení vysvětlované problematiky, která může být například v textové podobě hůře pochopitelná. Aplikace ovšem může být použita pro všechny činnosti, kde je potřeba jasně prezentovat vlastní nápad. Zároveň díky multimediálnímu zaměření může být využita i pro výuku a zábavu.

Aplikace je založena především na protokolu sítě XMPP. Díky otevřenosti a flexibilitě tohoto protokolu může aplikace nabízet požadované vlastnosti a dále být otevřená pro okolní svět (tedy ostatní programy stejného či podobného určení). Během vývoje se velmi osvědčila kombinace knihovny Qt 4 a aplikační platformy KDE 4 pro vývoj kvalitních desktopových aplikací. Tato platforma velmi zjednodušuje vývoj jak kvalitními nástroji, které jsou k dispozici, tak především kvalitně provedeným návrhem těchto knihoven z pohledu jejich aplikačního rozhraní. Aplikace navíc velmi dobře zapadá do prostředí KDE a uživatelé se nemusí učit zcela nové ovládání. Díky velkému rozsahu použitých postupů jsem si rozšířil vlastní znalosti obou knihoven do větší hloubky. Velká rozmanitost použitých technologií mi zároveň dala jak znalosti v problematice komunikace po síti Internet, tak v návrhu uživatelského rozhraní, které tuto komunikaci využívá. Díky využití otevřených standardů je aplikaci možné velmi dobře spravovat a především rozšiřovat. Dokáže taky kooperovat s ostatními produkty, které podporují otevřené standardy. Tak mohou vznikat další alternativy a v konkurenčním prostředí tlačit na další rozvoj projektu. Mezi plány pro budoucí rozvoj aplikace patří především vydání stabilní verze po otestování více uživateli a poté implementace real-time hlasové a video komunikace. Situace v této oblasti je složitější především z důvodů stále neschváleného standardu rozšíření protokolu XMPP Jingle. V přípravě je také rozšíření plochy o další multimediální a interaktivní objekty.

Velmi potěšující zprávou pro další vývoj aplikace je velký zájem komunity – jak vývojářů, tak především potenciálních uživatelů. Ukazuje se, že v (nejen) open source světě podobný nástroj citelně chybí a i přes nadějně pokusy o implementaci do stávajících IM klientů není žádný v takovém stavu, aby byl použitelný běžnými uživateli. Makneto je sice v době psaní

této technické zprávy ve fázi vývoje, ale během veřejných prezentací projektu (jak v rámci předmětů na FIT VUT v Brně, tak i při prezentaci zadavateli – společnosti Red Hat) byla přijata velmi pozitivně. Začíná se taky formovat komunita a v současné době na projektu spolupracují další dva vývojáři především jako testeři a výpomoc v oblasti webové prezentace a informací komunitě. Na přání zadavatele diplomové práce je dalším úkolem začlenění aplikace přímo do prostředí KDE. Otvírá se tak potenciální možnost rozšíření aplikace mezi další uživatele a především to může přilákat další vývojáře a celkově tedy přispět k rychlejšímu vývoji.

Tento projekt byl pro mne velmi přínosný a jsem rád, že jsem mohl komunitě přispět svojí prací. Během vývoje jsem se setkal i s potřebou vedení a prezentací projektu. Při vývoji jsem měl možnost spolupracovat s odborníky firmy Red Hat, kteří mi byli velmi nápomocni při návrhu a rozhodování o směru vývoje aplikace. V současné době kolem aplikace začíná vznikat vlastní komunita a věřím, že spousta uživatelů pomůže v jejich práci a usnadní jim vzdálenou komunikaci.

Literatura

- [1] DJ Adams. *Programming Jabber*. O'Reilly Media, 2002. ISBN 0-596-00202-5.
- [2] Edward C. Bailey. *Maximum RPM: Taking the Red Hat Package Manager to the Limit*. 1997. ISBN 1-888172-78-9.
- [3] Inc. Free Software Foundation. Gnu general public license, version 2.
<http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>, 1991.
- [4] Joonas Govenius. Svg whiteboardning.
<http://www.xmpp.org/extensions/inbox/wb.html>, 2006.
- [5] Mark Summerfield Jasmin Blanchette. *C++ GUI Programming with Qt 4*. Prentice Hall, 2006. ISBN 0131872494.
- [6] Eric Steven Raymond. Software release practice howto.
<http://en.tldp.org/HOWTO/Software-Release-Practice-HOWTO/index.html>, 2005.
- [7] Zack Rusin. Using kconfig xt.
<http://developer.kde.org/documentation/tutorials/kconfigxt/kconfigxt.html>, 2008.
- [8] Iain Shigeoka. *Instant Messaging in Java – The Jabber Protocols*. Manning Publicatins Co., 2002. ISBN 1-930110-46-4.
- [9] WWW stránky. Notateit - whiteboard software for tablets and pc's.
<http://www.notateit.com>, 2008.
- [10] WWW stránky. Xmpp standards foundation. <http://www.xmpp.org>, 2008.
- [11] Trolltech. Phonon overview.
<http://doc.trolltech.com/4.4rc1/phonon-overview.html>, 2008.
- [12] W3C. Scalable vector graphics (svg) 1.1 specification. <http://www.w3.org/TR/SVG/>, 2003.
- [13] Wikipedia. Open source software.
http://cs.wikipedia.org/wiki/Open_source_software, 2008.
- [14] David Yip. The inkboard protocol specification. <http://www.inkscape.org/>, 2007.

Dodatek A

The Makneto Handbook

Jaroslav Reznik
Revision 0.1 (2008-04-26)
Copyright ©2008

Makneto is shared whiteboard collaboration tool.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts.

A.1 Preface

A.1.1 Requirements

To run Makneto on your system, you have to meet these requirements:

- KDE 4 (kdebase and kdelibs)
- Qt libraries (version 4.3 and above)

Most of these items are distributed with your Linux distribution or can be found in repositories as packages. The package manager should meet these requirements automatically. Please refer to your distribution documentation.

A.2 Introduction

A.2.1 What is Makneto?

Makneto is shared whiteboard collaboration tool for KDE 4. It is designed to make easier sharing your ideas, drawings and multimedia content over network.

Main ideas

- Whiteboard drawing area
- Instant messaging

- Embedding rich multimedia and interactive content on whiteboard
- Voice/video chat

Makneto is based on XMPP technology (aka Jabber).

Makneto is still in state of development and some features are not implemented yet!

A.3 Getting started

To use Makneto you need to have Jabber ID. If you don't have any, please first register one. Refer to <http://www.jabber.org/im-services>.

Note: Makneto currently does not provide support for registering new account so you have to use some other XMPP client or web registration form.

A.3.1 Account settings

To set up your Jabber account login information, use Settings-Configure Makneto menu item. This will display Configuration dialog.

A.3.2 Go Online!

After setting up proper login data you can try to connect to service. Select Network tab (selected as default) and click on Online button (or other appropriate one).

Once you're online, your contacts will be fetched from the server and displayed in the Contact List. To start sharing board, just click their name and a Shared board session will appear. To draw something, use appropriate tool and your pointing device (mouse, tablet, ...).

A.4 Using Makneto

This chapter gives an overview of Makneto's basic features. We will look first at the contact list, where your contacts are displayed, and then at the Shared Board.

A.4.1 The Contact List

The Contact List appears when you select Contacts tab. You can begin new Shared Board session by right-click on selected contact and selecting New session menu item. You can add new contact a filer contacts by their presence and name. You can set your presence on The Network tab by clicking appropriate button.

A.4.2 The Shared board

The Shared Board session view is divided into two parts. One part for a chat-like communication and another one for a real-time drawing communication.

Drawing

Available tools for drawing are freehand, line, rectangle and ellipse. You can move, translate and remove any selected object. To select an object (or more objects) use Select tool from toolbar. Right click on selected object will show contextual menu. You can select stroke-width and filling color.

A.5 Configuring Makneto

To configure Makneto, look in the Settings menu.

A.5.1 The Configure Makneto Dialog

We briefly showed you how to add an account in Getting Started. On the Account page, you can set up your login information.

The Network tab are options:

- Jabber ID. Use your service Jabber ID or obtain new one.
- Password.
- Store Password. When this is checked, Makneto will store the password for you, so you don't have to enter it every time you connect to the IM service. If you want to limit access to the IM account you can leave this unchecked.

You can override default settings on the Connection tab.

The Connection tab options are:

- Host. Makneto will use manually entered hostname.
- Port. Default 5222.
- File transfer proxy.

Security options:

- Allow plain text login.
- Use SSL encryption.
- No SSL warnings.

A.6 Menu Entries

A.6.1 The File Menu

File-Quit (Ctrl+Q) Disconnects you from network and exits application.

A.6.2 The Settings Menu

Settings-View Menubar (Ctrl+M) Shows/hides the menu bar.

Settings-View Statusbar Shows/hides the status bar.

Settings-Full Screen Mode (Ctrl+F) Maximizes Makneto to fullscreen to gain some more place for the drawing area. You can combine it with Settings-View Menubar and hiding Sidebar.

Settings-Configure Shortcuts... Shows the KDE standard Configure Shortcuts dialog, where you can change keyboard shortcuts that work in the Contact List's windows.

Settings-Configure Toolbars... Displays the KDE standard Configure Toolbars dialog, which lets you customize the Contact List's toolbars.

Settings-Configure Makneto... Displays the Configure Makneto dialog.

A.6.3 The Help Menu

Help-Makneto Handbook (F1) Invokes the KDE Help system starting at the Makneto help pages. (this document).

Help-What's This? (Shift+F1) Changes the mouse cursor to a combination arrow and question mark. Clicking on items within Makneto will open a help window (if one exists for the particular item) explaining the item's function.

Help-Report Bug... Opens the Bug report dialog where you can report a bug or request a wishlist feature.

Help-About Makneto This will display version and author information.

Help-About KDE This displays the KDE version and other basic information.

A.7 Credits and Licences

Makneto is open source shared whiteboard tool under GNU General Public License.

Contributors:

- Jaroslav Řezník (rezzabuh AT gmail.com) : Original author, developer, project leader.
- Jozef Mlích (xmlich02 AT stud.fit.vutbr.cz) : Tester.
- Martin Hlahůlek (hlahulek AT merax.net): Tester, webhosting.

Many thanks to other open source developers of parts that Makneto relies on, especially Psi IM ones.

Thanks to Merax Networks for webhosting and free domain! Check theirs CMS!

This documentation is licensed under the terms of the GNU Free Documentation License.

This program is licensed under the terms of the GNU General Public License.

Dodatek B

SPEC soubor

```
# spec file for package makneto (Version 1.1.2)

# norootforbuild

Name:          makneto
BuildRequires: libkde4-devel
License:       GPL v2
Url:           http://www.makneto.org/
Group:         System/GUI/KDE
BuildRoot:     %{_tmppath}/%{name}-%{version}-build
Summary:       A shared board collaborative tool.
Version:       0.1
Release:       1
Packager:      Jaroslav Reznik <rezzabuh@gmail.com>
%define tar_version %version
Source:        makneto-%tar_version.tar.bz2

%description
Makneto is open source shared board collaborative tool.

%debug_package
%prep
%setup -n makneto-%tar_version -q

%build
    %cmake_kde4 -d build
    %make_jobs

%install
    cd build
    %makeinstall
    cd ..
    %find_lang %name
```

```
%post -p /sbin/ldconfig

%postun -p /sbin/ldconfig

%clean
rm -rf $RPM_BUILD_ROOT

%files -f %name.lang
%defattr(-,root,root)
%doc AUTHORS COPYING README
/usr/bin/makneto
/usr/share/applications/kde4/makneto.desktop
%doc %lang(en) /usr/share/doc/kde/HTML/en/makneto
/usr/share/applications/kde4/makneto.desktop
/usr/share/config.kcfg/makneto.kcfg
/usr/share/apps/makneto/maknetoui.rc
/usr/share/icons/hicolor/22x22/actions/maknetoonline.png

...

/usr/share/icons/hicolor/48x48/apps/makneto.png

%changelog
* Thu May 01 2008 rezzabuh@gmail.com
- Initial package (0.1)
```