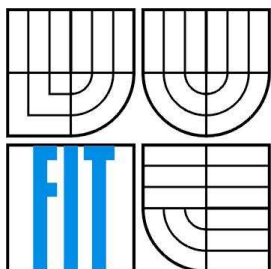




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

IMPLEMENTACE KLIENTA PRO ICQ IMPLEMENTATION OF KLIENT FOR ICQ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MILOSLAV RŮŽIČKA

VEDOUcí PRÁCE
SUPERVISOR

ING. ROMAN LUKÁŠ, PH.D.

BRNO 2007

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Růžička Miloslav**

Obor: Informační technologie

Téma: **Implementace klienta pro ICQ**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se podrobně s existujícími klienty pro ICQ.
2. Navrhněte vlastního klienta ICQ podle požadavků vedoucího BP.
3. Daného klienta naimplementujte.
4. Porovnejte přednosti a nedostatky Vámi naimplementovaného klienta ICQ s ostatními již existujícími klienty.

Literatura:

- Dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1) a 2)

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Lukáš Roman, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Miloslav Růžička**
Id studenta: 84165
Bytem: Pod Tržištěm 775, 675 31 Jemnice
Narozen: 24. 07. 1984, Třebíč
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Implementace klienta pro ICQ
Vedoucí/školitel VŠKP: Lukáš Roman, Ing., Ph.D.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....
David Š.

Autor

Abstrakt

Práce vysvětluje pojem instant messaging (IM), uvádí současné IM programy. Podává pohled na nejrozšířenější program v České republice pro instant messaging ICQ a na alternativní klienty. Práce dokumentuje návrh a vývoj ICQ klienta v prostředí Delphi.

Klíčová slova

ICQ, Delphi, Instant messaging, Oskar protokol, komponenta.

Abstract

Work explains conception instant messaging (IM), it features IM programs. Serves view of most widespread program in Czech Republic for instant messaging ICQ and on alternative clients. Work proves by evidence proposal and development ICQ client in environment Delphi.

Keywords

ICQ, Delphi, Instant messaging, Oscar protocol, component.

Citace

Miloslav Růžička: Implementace klienta pro ICQ, bakalářská práce, Brno, FIT VUT v Brně, 2007

Implementace klienta pro ICQ

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Romana Lukáše, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Miloslav Růžička
10.5.2007

Poděkování

Děkuji svému vedoucímu Ing. Romanu Lukášovi, Ph.D. za odborné vedení a cenné rady při návrhu programu. Dále děkuji svojí mamince, která mě po dobu realizace práce podporovala.

© Miloslav Růžička, 2007.

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

Obsah	1
Úvod	3
1 Instant messaging	4
1.1 Definice	4
1.2 Výhody	4
1.3 Nevýhody	4
1.4 Historie	4
2 IM klienti	5
2.1 Yahoo! Messenger	5
2.2 Windows Live Messenger	5
2.3 Google Talk	5
2.4 Skype	5
2.5 ICQ	6
2.5.1 ICQ v číslech	6
2.5.2 Základní pojmy spojené s IM a ICQ	6
2.5.3 Alternativní ICQ klienti	7
2.5.4 Zápory alternativních klientů	8
2.5.5 Klady alternativních klientů	8
3 Oscar protokol	10
4 Prostředí Delphi	11
5 Vlastní klient pro ICQ	12
5.1 Návrh klienta	12
5.1.1 Speciální funkce klienta	13
5.1.2 Konfigurační soubory	13
5.2 Implementace klienta	14
5.2.1 Popis tříd aplikace	15
5.2.2 Export/import historie zpráv	19
5.2.3 Přidávání kontaktů	19
6 Vlastní komponenty	20
6.1 Komponenta IcqMemo	20
6.1.1 Požadavky	20
6.1.2 Implementace	20
6.1.3 Problémy s implementací	21
6.2 Komponenta IcqMemoZ	22

6.2.1	Požadavky	22
6.2.2	Implementace	22
7	Závěr	24
	Literatura	25
	Seznam použitých zkratk a symbolů.....	26
	Seznam příloh	27

Úvod

Jednou z vlastností typické pro lidi je komunikace. Lidé mají už od pradávna potřebu si povídat, vyměňovat si zážitky nebo řešit problémy s osobami sobě blízkými. V době bez počítačů to byl samozřejmě osobní kontakt, při kterém si lidé popovídali. Po vynalezení papíru a se vznikem pošty se část této komunikace přesunula na psanou formu. Lidé spolu mohli komunikovat, i když jim vzdálenost mezi nimi nedovolovala osobní kontakt. Mělo to ale jeden zádrhel, a to čas potřebný k doručení zprávy od odesílatele k příjemci. S rozvojem výpočetní techniky, kdy se počítače dostaly i do běžných domácností, lidé začali používat elektronickou poštu. Čas potřebný pro doručení zprávy se zkrátil na několik sekund a pro doručení odpovědi je závislý čistě na adresátovi, až si zprávu vyzvedne a odepíše. Elektronická pošta má však nevýhodu v tom, že člověk píšící zprávu neví, jestli její příjemce sedí na druhé straně u PC a neví, kdy může očekávat odpověď. Tento problém řeší tzv. instant messaging (viz. definice níže), který uživatelům umožňuje sledovat stav kontaktů. Tato služba se těší stále větší oblibě.

Tato práce stručně popisuje vývoj instant messagingu, nejvíce rozšířené komunikační služby s klienty. Dále pak návrh vlastního klienta a jeho implementaci. V závěru práce srovnává tohoto klienta s profesionálními řešeními, ukazuje na jeho výhody, ale také na nevýhody.

V první kapitole se práce zabývá pojmem Instant messaging, v druhé IM klienty, ve třetí Oskar protokolem, ve čtvrté prostředím Delphi, pátá kapitola rozebírá návrh a implementaci vlastního klienta, šestá popisuje naprogramované komponenty a v sedmé shrnuje výsledky práce.

1 Instant messaging

1.1 Definice

Instant messaging je internetová služba umožňující svým uživatelům sledovat, kteří jejich přátelé jsou právě připojeni a dle potřeby jim posílat zprávy, chatovat, posílat soubory mezi uživateli a i jinak komunikovat. Hlavní výhoda oproti používání např. emailu spočívá v principu odesílání a přijímání zpráv v reálném čase. Jinými slovy zpráva je doručena ve velmi krátké době od odeslání (většinou v rámci stovek milisekund).

1.2 Výhody

Instant messaging zrychluje komunikaci a umožňuje snadnou spolupráci mezi více lidmi. Na rozdíl od emailu nebo telefonu druhá strana ví, zda je účastník k dispozici či nikoliv. Většina IM systémů umožňuje nastavit tzv. *away message*, tedy zprávu, která uvádí důvod, proč uživatel na druhé straně neodpovídá (je pryč). Na druhou stranu uživatele nikdo nenutí, aby na zprávy odpovídal ihned. Tímto způsobem se IM komunikace stává méně vyrušující než třeba telefon, a to je částečný důvod, proč je tento způsob komunikace stále více oblíben v obchodním prostředí. Instant messaging je ideální také pro rychlou výměnu internetových adres, kusů zdrojového kódu a dalších věcí.

1.3 Nevýhody

Programy pro instant messaging jsou většinou naprogramovány tak, aby komunikace procházela přes firewally a další bezpečnostní prvky na síti, proto se stávají potenciální hrozbou pro systém.

1.4 Historie

Prvním z volně dostupných programů široké veřejnosti a zároveň jedním z nejrozšířenějších programů pro instant messaging je ICQ izraelské firmy Mirabilis představený v roce 1996. Díky jeho velmi rychlému růstu se stal hrozbou pro komerční IM síť amerického poskytovatele internetu AOL Instant Messenger, který na to zareagoval zakoupením firmy Mirabilis. Během několika let bylo vyvinuto mnoho alternativních IM klientů (Yahoo Messenger, MSN Messenger, Excite, Ubique, či bezpečnější program Lotus Sametime od firmy IBM), každý s vlastním protokolem (pochopitelně navzájem nekompatibilním). Rozdíl mezi protokoly stírá otevřený XMPP protokol pro IM (potažmo Jabber).

Tato kapitola byla inspirována [8].

2 IM klienti

V této kapitole budou stručně popsány nejpoužívanější IM služby. Dále se zaměřím na ICQ, jeho popis, popis alternativních klientů, jejich výhod a nevýhod. Čerpáno z [5].

2.1 Yahoo! Messenger

Tuto komunikační službu provozuje internetový portál Yahoo!. Klient Yahoo! Messenger pracuje v komunikačních sítích Yahoo! a MSN, nabízí stejně jako ICQ přenos textu, hlasu i obrazu. Mezi další funkce patří například i přenášení souborů, konferenční hovory, internetový vyhledávač, RSS čtečka atd. Yahoo! Messenger není k dispozici v češtině, podobně jako ICQ zobrazuje v hlavním okně reklamní banner.

2.2 Windows Live Messenger

Windows Live Messenger je komunikační aplikace, která pracuje v sítích MSN a Yahoo! a může být propojena s e-mailovou schránkou na serveru Hotmail. Přenášet je možné nejen textové zprávy, ale i hlas, video a soubory. Windows Live Messenger umožňuje hrát hry po internetu, sdílet soubory i plochu vašeho počítače a obsahuje internetový vyhledávač. Je k dispozici i v češtině.

2.3 Google Talk

Komunikační aplikace Google Talk umožňuje textovou a hlasovou komunikaci po internetu. Google Talk nepřetéká funkcemi, na druhou stranu je však skvěle propojen s oblíbenou freemailovou službou Gmail a internetovým vyhledávačem Google a umožňuje například posílat hlasové e-maily (klasické e-maily s přílohou ve formě zvukového souboru). Google Talk se snadno ovládá, zatím ovšem není k dispozici v češtině.

2.4 Skype

Velmi rychle roste počet uživatelů sítě Skype, která umožňuje všechny běžné formy komunikace (text, hlas, obraz), přenášení souborů, konferenční hovory nebo chaty s více uživateli a mnoho dalšího. Síť Skype je navržena tak, aby ji nezastavila síťová zařízení jako proxy servery či firewally. Často vám tedy umožní komunikovat i v situacích, kdy není možné spojení pomocí ostatních komunikačních sítí. Velmi zajímavá je služba Skype Out, která umožňuje volání z počítače na běžná telefonní čísla za velmi příznivých cenových podmínek. Pro klienta Skype je k dispozici celá řada

doplňků, jako jsou telefonní záznamníky a další aplikace. Samozřejmostí je plně počeštěné uživatelské rozhraní.

2.5 ICQ

ICQ je protokol pro instant messaging a zároveň komunikační software vyrobený izraelskou firmou Mirabilis a poprvé uvedený v roce 1996. Nynější vlastník je firma AOL, která v roce 1998 koupila firmu Mirabilis za 394 milionů dolarů. ICQ se vyslovuje jako „ajsíkjú“, což v angličtině zní jako „I seek you“, neboli „hledám tě“. Síť ICQ umožňuje zasílat textové zprávy, URL, celé soubory, navštívenky, umožňuje chat, odesílání SMS s denním limitem, různé hry a další. Nevýhodou je, že komunikace není šifrována. Bez instalace oficiálního programu se lze na ICQ připojit také pomocí webového rozhraní ICQ2Go (ve Flashi a Javě), nebo pomocí alternativních klientů. Uživatelé ICQ jsou číslování pomocí čísla UIN (Universal Internet Number nebo Unified Identification Number), jež je pro každého uživatele unikátní. Noví uživatelé nyní dostávají UIN přes 300 000 000. Nízká čísla vydražil eBay od uživatelů, jež se zaregistrovali velmi brzo. Od roku 2000 se můžou uživatelé ICQ a AIM vkládat navzájem do svých kontaktů bez pomoci jiného klienta.

Použitá literatura [7].

2.5.1 ICQ v číslech

- více jak 370 000 000 stažení ze serveru download.com,
- průměrný uživatel je připojený k síti ICQ víc než 5 hodin denně,
- denní traffic: 400 000 000 zpráv,
- denně přihlášených uživatelů 8 000 000,
- aktivních uživatelů 20 000 000.

K těmto úctyhodným číslům musíme dodat, že ICQ přes internet denně přepumpuje stovky miliónů reklamních bannerů. Bohužel, spolu s běžným obsahem (textem a soubory) se pomocí ICQ přenáší stále více virů a jiné škodlivé kódy.

2.5.2 Základní pojmy spojené s IM a ICQ

- **UIN (Unified Identification Number)** - unikátní číslo každého registrovaného uživatele sítě ICQ. Díky tomu, že toto číslo stále roste, je počet uživatelů teoreticky neomezený. Pro přehlednost se v tzv. „kontakt listu“ zobrazují také přezdívky / nicky jednotlivých uživatelů - narozdíl od UIN nejsou nicky unikátní (a tak současně připojených uživatelů "Michal" můžeme na ICQ najít několik).

- **Protokol** - je společný "jazyk", pomocí kterého se komunikátory spolu domlouvají. Počítá se také s komunikací s centrálním serverem (nebo několika servery), které zajišťují přihlášení k síti. Pokud nějaký komunikátor zvládne několik různých protokolů, může se většinou připojit i na několik IM sítí. A právě tento fakt umožňuje být na síti ICQ i bez firemního ICQ klienta.
- **Klient** - klientský program se obvykle instaluje na počítač uživatele (existují ale také internetové klientské programy napsané ve Flashi nebo v Javě). Klient provádí přihlášení (připojuje se na server IM sítě), eviduje kontakty a umožňuje posílání a příjem zpráv.
- **Adware** - Firma Mirabilis (a nyní AOL) se rozhodla, že k bezplatné síti ICQ připojí reklamní systém, který bude pokrývat náklady a zároveň bude zdrojem případného výnosu. Bannery se zobrazují pod „kontakt listem“ a pod každým komunikačním okénkem (ICQ Message Window Banner).
- **Status** - Status je u každého kontaktu v ICQ reprezentován malou ikonou reprezentující stav uživatele (online, offline, pryč, nerušit...). Typickou ikonou u ICQ je kytička - zelená, pokud je kontakt online; červená, pokud je offline.
- **Server** - v IM síti zprostředkovávají servery přihlášení k síti a zároveň monitorují status uživatele (online, away, invisible...). Komunikace připojených klientů probíhá přímým způsobem - v některých zvláštních případech (příjemce zprávy je např. offline) servery zprávy archivují a doručují.

S menšími úpravami převzato z [3].

2.5.3 Alternativní ICQ klienti

2.5.3.1 Miranda IM

Miranda je více-protokolový instant messenger (ICQ, MSN, Jabber ...).

Na první pohled působí jako zastaralý klient. Málo barev, relativně hranaté okraje, žádné ovládací prvky. Tím však výčet nevýhod končí. Možná by se našel ještě jeden, a to problémy s některými pluginy. To je však asi poslední nevýhoda. Miranda je jinak absolutně nejméně náročný program, co se hardwaru týče. Její jednoduchý vzhled je spíše výhodou než nevýhodou. Zobrazuje jen podstatné věci a neotravuje uživatele zbytečnostmi, jako jsou např. reklamy. Tou největší výhodou jsou ale pluginy. Uživatel si tak může svoji instalaci přetvořit k obrazu svému, od přidání smajlíků, odesílání sms, budíku, psaní poznámek, po nejrůznější statistiky. Jak již bylo uvedeno, Miranda umí spoustu nadstandardních věcí, které si ale uživatel musí doinstalovat a nakonfigurovat, z toho tedy

vyplývá, že tento klient je spíše pro zkušenější uživatele PC a ne pro začátečníky. Já osobně jsem měl problém s kódováním diakritiky a s instalací pluginu pro smajlíky.

Klient je ve verzi pro Windows.

2.5.3.2 SIM

Tento klient pochází z Ruska. Tvůrci si zřejmě řekli, že v jednoduchosti je krása a celý design tomuto podřídili. Klient má hranaté tvary, nepříliš dobře zvolené barvy, ikony z prvních verzí originálního klienta. I tento klient podporuje pluginy (je jich ale menší množství než pro Mirandu). Největší přednost tohoto klienta však spočívá v možnosti být přihlášen k více účtům na stejném protokolu. Mohu být tedy přihlášen jak já, tak třeba i můj bratr, pokud zrovna oba potřebujeme být ve stejnou chvíli přihlášení.

Existuje verze pro Windows i pro Linux.

2.5.3.3 QIP

Na rozdíl od Mirandy má příjemnější uživatelské rozhraní a jednodušší ovládání. Klient podporuje skiny pro změnu vzhledu. Odlišností od ostatních klientů je to, že komunikace pro všechny klienty probíhá v jednom okně, ve kterém se pomocí záložek přepíná mezi kontakty. Další nadstandardní funkcí je antispamový filtr, který odfiltruje nežádané zprávy. Systémové nároky jsou větší než u Mirandy, ale QIP v základní instalaci obsahuje mnohem více možností a funkcí než Miranda.

Klient je ve verzi pro Windows.

2.5.3.4 Trillian

Trillian je více-protokolový klient, který se dokáže připojit na nejrozšířenější IM sítě (ICQ, AOL, MSN a Yahoo!). Je velice propracovaný. Nevýhodou je pomalejší překreslování grafiky.

Klient pro Windows, Mac OS.

2.5.4 Zápory alternativních klientů

Asi největší nevýhodou alternativních klientů je jejich nefunkčnost při změně protokolu a jejich nutný update. Dále špatně nebo vůbec funguje posílání souborů. Také neobsahují hry, které jsou pro jednu skupinu uživatelů podstatné, ale pro druhou skupinu to je spíše výhoda, neboť s tím souvisí menší nároky na hardware počítače.

2.5.5 Klady alternativních klientů

Menší systémové nároky, i když v dnešní době už možná nerelevantní argument, jelikož výkon počítačů a především množství operační paměti RAM v osobních počítačích velmi rychle stoupá, takže tento neduh nevíce vyčítaný originálnímu klientu nemusí být brán v potaz. Na druhou stranu, i přesto by se měly vyvíjet programy, které zbytečně nezabírají více paměti, než je nutné.

Různá vylepšení od statistik připojení uživatelů v seznamu kontaktů, přes příjemnější uživatelské rozhraní, po další funkce přímo nesouvisející s komunikací nebo zprávou kontaktů, jako je například budík. U klientů podporujících pluginy je to také možnost vytvořit si klienta přesně sobě na míru.

Nezobrazování otravujících reklamy je další nespornou výhodou těchto programků, která je nejenom otravující svou vizuální podobou, ale také tím, že zvyšuje množství přenášených dat po síti.

3 Oscar protokol

Jedná se o binární TCP protokol, jehož název OSCAR symbolizuje slova „Open System for CommunicAtion in Realtime“, po volném překladu „Otevřený systém pro komunikaci v reálném čase“. Navzdory slovíčku „open“ v názvu je specifikace protokolu uzavřená. V současné době využívají OSKAR protokol dva klienti firmy AOL pro instant messaging: ICQ a AIM. Firma AOL v minulosti vyvinula velké úsilí, aby se konkurentům jako je Microsoft, Jabber a Cerulean Studios nepodařilo naimplementovat kompatibilní software pro posílání zpráv. V roce 2002 poskytla firma AOL firmě Apple Computer vlastní knihovnu, která umožnila kompatibilitu mezi produkty AOL a programem iChat firmy Apple Computer. Na oplátku si mohli klienti AOL vytvořit email `login@mac.com`.

Další zpřístupnění protokolu třetím stranám se nekonalo a dostupné specifikace protokolu na internetu vznikly po tzv. reverse-engineeringu (což v tomto případě ve zkratce znamená, že se odchytne komunikace originálního klienta a zpětně se dešifruje). Což samozřejmě způsobuje ne stoprocentní kompatibilitu s protokolem.

Firmě AOL tak stačí menší změna protokolu a alternativní klienti přestávají fungovat. Tohoto jednání se firma dopouští z toho důvodu, protože vývoj a provoz ICQ je hrazen z reklamy distribuované společně s originálním klientem a tedy alternativní klienti jsou nežádoucím jevem.

6. března 2006 AOL uvolnilo AIM SDK a spustilo vývojářský web pomáhající vývojářům vytvářet pluginy pro AIM. Specifikace OSKAR protokolu však uvolněna nebyla.

Inspirováno z [6].

4 Prostředí Delphi

Delphi je integrované grafické vývojové prostředí firmy Borland, určené pro tvorbu aplikací na platformě MS Windows v jazyce Object Pascal (objektové nástavbě Pascal). Obsahuje systém RAD (Rapid Application Development), který umožňuje vizuální návrh grafického uživatelského rozhraní, na jehož základě je automaticky vytvářena kostra zdrojového kódu, což výrazně urychluje vývojový cyklus. Programování v Delphi je z velké části založeno na použití komponent. Komponenta je malý program (balíček funkcí), který vykonává určitou činnost (například zobrazuje text nebo obrázky, přehrává multimédia, komunikuje s databází, zprostředkovává FTP přenos, atd.).

Velkou předností Delphi proti některým konkurenčním produktům jsou již zmiňované knihovny komponent, které jsou jejich součástí (např. VCL, CLX, Indy ...). Dodávané komponenty významně usnadňují tvorbu aplikací. Další komponenty lze stáhnout z internetu (některé jsou zadarmo, některé se musí koupit). V Delphi lze vytvářet vlastní komponenty.

Snaha firmy o přenositelnost na platformu Linux vedla k vývoji prostředí Kylix. Je to integrované vývojové prostředí, určené pro vývoj aplikací v operačním systému GNU/Linux. Mezi produkty Kylix a Delphi je zajištěna částečná (je třeba používat jen vybranou množinu komponent) přenositelnost kódu. Vývoj Kylixu byl již ukončen a neočekávají se žádné další verze.

5 Vlastní klient pro ICQ

5.1 Návrh klienta

Co se týká vzhledu, klient nepřináší žádné novinky a zapadá do nastoleného trendu mezi klienty pro IM, jelikož nezvyklý vzhled nebo ovládání programu by mohlo způsobilo nežádoucí zmatení uživatele.

Klient bude splňovat základní funkčnost, která se od klienta pro IM očekává, tedy bude umět načíst seznam kontaktů ze serveru, přijímat a odesílat zprávy, měnit status přihlášeného uživatele a také bude umět vyhledávání kontaktů podle ICQ čísla, podle jména a příjmení a podle nicku. Nalezený kontakt si bude moci uživatel přidat do svého seznamu kontaktů s tím, že se toto přidání projeví také na straně serveru a při příštím přihlášení i přes jiného klienta bude kontakt načten spolu s ostatními kontakty.

Program bude svoji činnost signalizovat ikonou v tray. Jak hlavní okno aplikace, tak i zástupce aplikace na hlavním panelu půjdou úplně skryt, aby aplikace nepřekážela jak na ploše, tak také na hlavním panelu, protože se předpokládá její nepřetržitý provoz od spuštění počítače po jeho vypnutí. Vzhledem (resp. barvou) ikony v tray bude zobrazován aktuální status uživatele.

V hlavním okně bude zobrazen seznam kontaktů s možností vypínání a zapínání zobrazení skupin. Každý kontakt bude ikonou informovat uživatele o stavu kontaktu.

Pro komunikaci s každým kontaktem bude zvláštní formulář, jelikož mi přijde toto řešení lepší než jedno okno pro všechny kontakty se záložkami pro každý kontakt. Uživatel bude mít možnost vkládat do zprávy grafické smajlíky a taktéž v příchozích zprávách se smajlíky budou zobrazovat v grafické podobě a ne jen v jakési textové náhradě. Příchozí a odeslané zprávy se budou ukládat do historie a v této historii bude uživateli umožněno vyhledávat a také mu bude umožněno na zobrazené zprávy aplikovat filtr (např. jen přijaté, nebo jen zprávy spadající do určitého měsíce atd.).

Nastavení bude umožňovat:

- zapnutí/vypnutí zvuku
- volba, která nastaví, které kontakty se budou zobrazovat (např. jen kontakty se statusem online apod.)
- zobrazení skupin
- průhlednost hlavního formuláře
- hlavní formulář zobrazen nad ostatními okny
- automaticky zobrazená zpráva, když je status uživatele nastaven na pryč
- ukládání přijatých a odeslaných zpráv

5.1.1 Speciální funkce klienta

Klient bude umožňovat „pozdní odeslání zpráv“, což znamená, že si uživatel napíše zprávu a místo toho, aby tato zpráva byla odeslána ihned, uživatel zadá ještě čas, ve kterém se má zpráva odeslat a ve stanovenou dobu bude zpráva automaticky odeslána.

Klient bude sledovat stav přihlášení o kontaktech v seznamu kontaktů a bude tuto historii přehledně zobrazovat pomocí grafu, a to 7 dní nazpět.

Další vymožeností je integrovaný budík. Uživatel má možnost si zapsat na určité datum připomínku a klient ho ve stanovený čas upozorní vizuálně a podle nastavení i za pomoci zvuku.

Klient bude umožňovat import a export historie zpráv a také bude umět importovat historii zpráv z originálního klienta verze 5.1 .

Klient nabízí nastavit si upozornění, pokud konkrétní kontakt přejde do stavu online. Originální klient a ostatní alternativní klienti nabízí také možnost upozornění na online status kontaktu, ale nastavuje se globálně pro všechny kontakty.

Při odesílání zprávy je možno zprávu napsat i delší, než je limit pro odesílané zprávy, klient zprávu rozdělí a odešle po částech.

5.1.2 Konfigurační soubory

U aplikace, kterou tato práce popisuje, se předpokládá, že ji bude používat více lidí na jednom počítači. Z tohoto důvodu je pro každého uživatele založena vlastní složka a v ní jeho vlastní konfigurační soubor. Formát souboru byl zvolen xml, jelikož jsou pro něj dostupné parsery. Klasicky (v dnešní době však již na ústupu) uložená konfigurace v ini-souboru by pro potřeby tohoto programu nevyhovovala. Aplikace umožňuje i bez přihlášení k síti používat integrovaný budík a jelikož tedy program nezná UIN uživatele, potažmo kde najde soubor s uloženými budíky, je pro budík v nepřihlášeném stavu použit globální soubor. Do globálního konfiguračního souboru (jedná se o jiný soubor, než ten pro budík) se také ukládají UIN čísla uživatelů po přihlášení k síti. Toho využívá přihlašovací formulář, který si při svém vytvoření načte tyto čísla a nabídne je uživateli, který tak své UIN musí vyplnit jenom jednou při prvním použití aplikace.

5.1.2.1 Soubory s historií

Jako formát souborů na ukládání historie byl zvolen textový soubor. Každý záznam je uložen na jednom řádku. Pokud příchozí nebo odesílaná zpráva obsahuje znaky pro odřádkování, jsou pro potřeby historie tyto znaky nahrazeny mezerou. Ukládá se datum a čas odeslání resp. přijetí zprávy, typ zprávy (přijatá / odeslaná) a vlastní text zprávy. Pro ukládání historie každého kontaktu je vytvořen zvláštní soubor. Název souboru je odvozen od UIN kontaktu a přidáním „_historie“ s koncovkou txt.

Příklad názvu souboru: 240845273_historie.txt

Příklad záznamu: *4.4.2007;20:53:32;Prichozi;to sem rad ze te tu vidim*

5.1.2.2 Soubory pro ukládání historie připojení kontaktu k síti

Formát těchto souborů byl zvolen opět textový. Pro každý kontakt je vytvořen vlastní soubor. Každý záznam je uložen na jeden řádek, ukládá se start klienta, čas a datum se stavem při změně a čas s datem při ukončení klienta. Název souboru je odvozen od UIN kontaktu a přidáním „_statushis“ s koncovkou dat.

Příklad názvu souboru: *240845273_statushis.dat*

Příklad záznamu: *online:18.4.2007 22:56:29*

5.1.2.3 Soubory pro ukládání budíku

Formát souboru je opět textový. Na každém řádku je uložena jedna upomínka. Ukládá se čas a datum v unix-formátu, typ upomínky a text zprávy. Čas se při ukládání převádí na typ Unix, který vyjadřuje počet sekund od 1.1.1970. Název souboru je „budik.txt“

Příklad záznamu: *1177874747;Se zvukem;Zavolat domu*

Strategie s ukládáním u tohoto souboru byla zvolena podobně jako u konfiguračních souborů. Pokud uživatel není přihlášen, je funkce budík přístupná, ale ukládání upomínek probíhá do globálního souboru. Po přihlášení má každý uživatel svůj soubor s upomínkami.

5.1.2.4 Shrnutí

Jelikož byly formáty souborů zvoleny (až na jednu výjimku) jako čistě textové, je potřeba při ukládání např. textu upomínky nebo textu zprávy odstranit z těchto textů znaky pro odřádkování, jinak by byla porušena konzistence souborů. Program to řeší tak, že znaky pro konce řádků nahradí mezerou. Oproti této nevýhodě přináší textový soubor výhodu ve snadném čtení dat. Nejsou potřeba žádné zvláštní techniky, stačí naprosto jednoduchý kód.

5.2 Implementace klienta

Pro komunikaci s ICQ-serverem jsou dvě možnosti.

První spočívá ve vlastní implementaci komunikace se serverem. Jelikož je však ICQ komerční software a není uvolněna specifikace protokolu, tato možnost by byla velice špatně realizovatelná. Musel bych odchyťovat komunikaci originálního klienta s ICQ-serverem a „odpozorovat“ specifikaci protokolu. Tato možnost by vyžadovala hlubší znalosti o protokolech a sítích a také by vyžadovala poněkud časově náročnější tvůrčí činnost, čímž by neúměrně stoupla obtížnost implementace klienta.

Druhá možnost spočívá v použití knihovny nebo komponenty pro komunikaci s ICQ-serverem, kterou používají alternativní klienti.

Pro tento projekt byla zvolena druhá možnost, a to komponenta TICQClient pro prostředí Delphi. Tato komponenta je volně ke stažení na adrese <<http://www.cobans.net>>. Bohužel byl ukončen její vývoj, což s sebou přináší několik problémů. Nebudou žádné nové verze, které by přinesly nové funkce. Do značné míry je prodloužen časový interval od změny protokolu po vydání opravy. Další nepříjemnou věcí je, že některé funkce komponenty přestávají fungovat, jako např. posílání souborů, špatně funguje funkce na převod přijaté zprávy ve formátu RTF do čistého textu a také se objevují jisté problémy s autorizací při přidávání kontaktů do seznamu kontaktů. Chybu funkce, která dekóduje text z přijaté zprávy ve formátu RTF do „čistého“ textu (dojde k převedení jen prvního řádku textu) klient řeší tak, že přijatou zprávu zobrazí do komponenty RichEdit na „logovacím formuláři“, tato komponenta má zapnuté dekódování formátu RTF na „čistý“ text, a tak příkazy na přiřazení zprávy do komponenty a opětovné přečtení textu z komponenty převedou zprávu z RTF do „čistého“ textu. Bohužel se někdy stane, že text převeden není (důvod se mi nepodařilo zjistit), takže je na text ještě aplikována funkce, která se pokusí text dekódovat.

Na druhou stranu se s touto komponentou velice dobře pracuje. Stačí ji umístit na formulář, vyplnit vlastnosti *UIN* a *Password* a zavolat metodu *Login* a komponenta se pokusí navázat spojení se serverem. Při úspěchu se vyvolá událost *OnLogin*, při neúspěchu událost *OnError*, kde je pomocí parametrů specifikován důvod neúspěchu. Další metody a události slouží k dalším činnostem klienta, jako např. metoda *SendMessage* s parametrem UIN kontaktu, komu je zpráva určena a textem zprávy a událost *OnMessageRecv* pro přijetí zprávy opět s parametry UIN kontaktu, od kterého zpráva přichází, a textem zprávy.

Vyjmenované neduhy by se daly odstranit zapojením se do vývoje komponenty a případně nefunkčnosti opravit. Komponenta je však dodávána jenom s popisem funkcí a událostí a ne s implementační dokumentací a jelikož počet řádků zdrojových kódů této komponenty se pohybuje kolem 10 tisíc, bylo by to velice obtížné a časově náročné.

5.2.1 Popis tříd aplikace

Třída TKontakty v jednotce ICQ_Kontakty.pas

Třída uchovává seznam kontaktů a seznam skupin pro další funkce programu. O kontaktu uchovává UIN, přezdívku, unikátní číslo kontaktu, číslo skupiny, status, ukazatel na formulář pro psaní zpráv, příznak, zda je kontakt na serveru a příznak pro upozornění na online status kontaktu. O skupině třída uchovává unikátní číselný identifikátor, název skupiny a počet kontaktů v této skupině.

Vnitřně je seznam kontaktů implementován lineárním spojovým seznamem, kontakt je přidáván na začátek tohoto seznamu, což je nejméně časově náročné (jelikož třída uchovává jenom ukazatel na první prvek seznamu). Seznam skupin je řešen dynamickým polem, při přidávání skupin je toto pole automaticky zvětšeno na aktuální počet skupin (včetně právě přidané skupiny).

Třída TStatusKontakty v jednotce Historie_Status_Kontaktu.pas

Slouží k ukládání historie připojení jednotlivých kontaktů. Třída ukládá historii posledních 7 dnů, kdy byl klient spuštěn. Z uložených dat metoda této třídy vykreslí graf a historie připojení kontaktu tak může být přehledně zobrazena. Třída udržuje vnitřní seznam kontaktů (vnitřně implementován dynamickým polem), o kterém si uchovává UIN a *StringList*, do kterého zaznamenává historii statusu. Poskytuje metody pro přidávání kontaktu, o kterém má být veden záznam. Po přidání kontaktu je do historie statusu (*StringListu*) přidána položka, která říká, že od tohoto data se začíná zaznamenávat historie statusů. V metodě *destroy* třídy jsou záznamy zkráceny na posledních 6 dní a jsou uloženy do příslušného souboru. Formát tohoto souboru je popsán v kapitole 5.1.2.2 .

Třída TUserInfoForm v jednotce UserInfo.pas

Třída slouží k zobrazení informací o kontaktu jako jsou jméno, příjmení, datum narození, pomocí třídy TStatusKontakty zobrazuje historii připojení atd. . Metodou *Init* se jí předá UIN kontaktu, o kterém má zobrazit informace z ICQ serveru a graf s historií připojení.

Třída TFormRename v jednotce Unit_Rename.pas

Třída obsahuje formulář pro přejmenování kontaktu v seznamu kontaktů. Formulář je zobrazen modálně. Pokud uživatel zadal nové jméno a stiskl tlačítko OK, formulář, který toto dialogové okno zobrazil (form1), si z public proměnné přečte nové jméno a pokusí se přejmenovat kontakt s tím, že se změna promítne také na serveru. O úspěchu či případném neúspěchu je uživatel informován informační hláškou.

Třída TFormBudik v jednotce Unit_budik.pas

Implementuje formulář pro nastavení připomínek a jejich zobrazení (pomocí třídy TFormAlarm v jednotce Unit_Alarm). Uložené připomínky jsou ukládány a při spuštění znova načítány ze souboru, takže uživatel nepřijde o nastavené připomínky ani když aplikaci ukončí. Formát souboru je popsán v kapitole 5.1.2.3 . Zprávy jsou při načtení uloženy do vnitřního bufferu (*StringList*) a zobrazeny do komponenty *ListView* na formuláři. Ve vnitřním bufferu je čas ve formátu unix (tedy počet sekund od 1.1.1970) z toho důvodu, že upomínky jsou v bufferu uloženy setříděné a setřídít čísla je jednodušší, než setřídovat řetězce s datem a časem. Na formuláři je položena komponenta *Timer*, pomocí které je každých 300 ms kontrolován aktuální čas s časem upomínky.

Třída TFormAlarm v jednotce Unit_Alarm.pas

Třída zobrazuje připomínku ve formě malého formuláře. Podle délky textu upraví svoje rozměry a efektně najede na plochu v pravém dolním rohu obrazovky.

Třída TForm9 v jednotce Unit9.pas

Implementuje formulář pro zobrazení informačních zpráv uživateli. Podle délky textu upraví (zvětší) svoje rozměry. Zobrazí text a ikonu symbolizující typ zprávy (chyba, varování, informace). Nahrazuje MessageDlg z Delphi, který by narušil jednotný vzhled aplikace.

Třída TForm8 v jednotce Unit8Options.pas

Implementuje formulář pro nastavení klienta. Nastavení ukládá do konfiguračního souboru a také po spuštění načítá z konfiguračního souboru. Formát souboru a strategie ukládání je popsána v kapitole 5.1.2 . Umožňuje uživateli zapnout zvuk, zobrazit kontakty do příslušných skupin, zobrazit jen některé kontakty, podle aktuálního statusu nastavit „away message“, nastavit ukládání příchozích a odeslaných zpráv, nastavit průhlednost hlavního formuláře s kontakty a jestli má být tento formulář zobrazen nad hlavními okny. Přes tento formulář se také provádí import a export vlastní historie zpráv a import historie zpráv z originálního klienta verze 5.1 .

Třída TForm6 v jednotce Unit6.pas

Zobrazuje formulář pro výběr a následné vložení smajlíka do zprávy. Smajlíci byli zkopírováni z originálního klienta.

Třída TForm5 v jednotce Unit5.pas

Zobrazuje formulář pro přidání kontaktu do seznamu kontaktů. Uživatel se na tento formulář dostane přes formulář pro vyhledávání kontaktů, může si změnit jméno kontaktu a případně skupinu, do které bude kontakt patřit. Skupinu může zvolit jen takovou, která již existuje. Přidávání nových skupin nebylo implementováno. Po stisknutí tlačítka OK je kontakt přidán. Aby se změny projeví i na serveru, musí uživatel použít volbu z hlavního menu „ update kontaktlistu na serveru“.

Třída TForm4 v jednotce Unit4.pas

Implementuje formulář pro změnu hesla. Uživatel je požádán o 1x vyplnění starého hesla a 2x vyplnění nového hesla. O úspěchu či neúspěchu je uživatel informován informačním oknem.

Třída TForm3 v jednotce Unit3.pas

Implementuje formulář pro komunikaci s kontaktem. Uživatel má možnost si napsat zprávu, vložit do ní smajlíka a odeslat. Odeslání zprávy je provedeno okamžitě nebo pokud uživatel zvolí „odeslat později ...“, zobrazí se mu formulář pro vyplnění času, kdy si přeje zprávu odeslat.

Na tomto formuláři se nachází ovládací prvek v podobě malé ikony pro spuštění formuláře s historií zpráv s tímto kontaktem. Při vytváření, resp. rušení formuláře je načítána, resp. ukládána pozice formuláře do konfiguračního souboru. Program si tak pamatuje poslední pozici formuláře na obrazovce a při dalším spuštění je formulář zobrazen na stejném místě.

Třída TForm2 v jednotce Unit2.pas

Implementuje formulář pro přihlášení. Uživatel zde vyplní svoje UIN a heslo a přihlásí se k serveru. V případě úspěšného přihlášení je UIN uloženo do konfiguračního souboru (jedná se o globální konfigurační soubor pro všechny kontakty), aby ho uživatel příště již nemusel ručně vyplňovat.

Třída TForm7 v jednotce Search.pas

Implementuje formulář pro vyhledávání kontaktů. Uživatel může vyhledávat podle UIN kontaktu nebo podle jména, příjmení a nicku kontaktu. Nalezený kontakt si může uživatel přidat do seznamu kontaktů.

Třída TFormHistorie v jednotce Historie.pas

Třída slouží k zobrazení historie zpráv mezi uživatelem a kontaktem ze seznamu kontaktů. Uživatel může na zobrazené zprávy aplikovat filtr s možnostmi zobrazit jen přijaté/odeslané/všechny zprávy spadající do určitého časového intervalu, či zprávy náležící do konkrétního data. Filtry se dají kombinovat. V zobrazených zprávách je umožněno vyhledávat.

Historie je ukládána do souboru, formát popisuje kapitola 5.1.2.1 .

Jednotka mXML

Tato jednotka poskytuje metody pro práci s xml souborem. Byla vytvořena pro usnadnění čtení a zapisování konfiguračních údajů, jelikož klient pro ukládání historie používá soubor ve formátu xml.

Třída TFormAutoZpravy v jednotce Unit_AutoZpravy.pas

Třída implementuje formulář pro zobrazení naplánovaných zpráv k automatickému odeslání. Uživatel může přes tento formulář naplánované zprávy zrušit nebo odeslat ihned, naplánovat je však musí přes formulář pro odesílání zpráv. Při ukončení aplikace klient upozorní na případné zprávy ve frontě, umožní zrušit ukončení aplikace, tyto zprávy ale neukládá, takže s ukončením aplikace dochází k jejich ztrátě.

Třída TFormCas v jednotce UnitCas.pas

Tato třída implementuje formulář sloužící k zadání času, ve kterém má dojít k odeslání naplánované zprávy. Pokud se uživatel snaží zadat menší čas, než je aktuální (čas do minulosti), je zobrazeno varování a program čeká na nové zadání času. Při správném zadání času je na komponentu, která slouží k zobrazení komunikace, vypsána informační hláška a zpráva je naplánována k automatickému odeslání.

Třída TForm1 v jednotce Unit1.pas

Třída implementuje hlavní okno aplikace, kde jsou zobrazeny kontakty. Uživatel si přes tento formulář může změnit status, odeslat zprávu kontaktu, zobrazit formulář s nastavením atd. Na tomto formuláři je také umístěna komponenta ICQClient, která zprostředkovává komunikaci s ICQ-serverem.

5.2.2 Export/import historie zpráv

Klient podporuje import a export historie zpráv. Formát souboru je textový, každý záznam má stejnou strukturu jako záznamy pro každý kontakt. Soubor s exportovanou historií je spojením všech souborů s historií pro každý kontakt, jen před každou částí od konkrétního kontaktu je značka, komu následující historie patří, ve tvaru zavináče a UIN (např. @209408495). Algoritmus tak jenom čte soubor po řádcích a pokud je první znak zavináč, další znaky do konce řádku jsou UIN kontaktu, jsou nastaveny příslušné proměnné a následující řádky (až po další zavináč na první pozici na řádku nebo do konce souboru), jsou ukládány do lokální historie zpráv ke kontaktu.

Originální klient verze 5.1 ukládá historii zpráv do XML souborů. Každému kontaktu je na každý měsíc vytvořen zvláštní soubor. Klient při importování historie načítá zprávy z těchto XML souborů a ukládá je do souborů jednotlivým kontaktům. Postup je takovýto: Klient si do *StringListu* pro daný kontakt načte „svoji“ historii zpráv, pak si do *StringListu* přidá záznamy z XML souborů, seřídí tento *StringList* podle data a času a uloží ho (jednotlivé záznamy) do „svého“ souboru s historií pro daný kontakt.

5.2.3 Přidávání kontaktů

Bohužel při přidání kontaktu do seznamu kontaktů se může objevit chybová hláška s tím, že tento kontakt neautorizoval (nesouhlasil) s přidáním kontaktu, i když ve skutečnosti byl souhlas udělen. Pokud je přidání kontaktu do seznamu kontaktů inicializováno od kontaktu na druhé straně, tedy kamarád si přidá mě do seznamu kontaktů a já pak jeho (testováno s originálním klientem 5.1), výše zmíněná chyba se neobjeví a přidání kontaktu proběhne bez problémů.

6 Vlastní komponenty

Pro tuto aplikaci bylo nutné naprogramovat dvě komponenty. Jedná se o komponenty pro psaní textu s vkládáním smajlíků. Pro všechny ostatní funkce a vlastnosti programu (až na jednu výjimku, jelikož byla doinstalována komponenta pro zobrazení ikony v tray) byly použity standardní komponenty dodávané s prostředím Delphi 7.

6.1 Komponenta *IcqMemo*

6.1.1 Požadavky

Komponenta bude umožňovat vkládání znaků a smajlíků, označování textu do bloku a jeho kopírování do schránky, vkládání textu ze schránky. V podstatě se bude jednat o standardní komponentu pro psaní textu s rozšířením o smajlíky.

6.1.2 Implementace

Jako předek této komponenty musí být zvolena třída, která disponuje událostmi pro interakci s klávesnicí a pro změny fokusu. Tyto vlastnosti má třída *TWinControl*. Za předka komponenty byla však zvolena třída *TCustomControl*, která je uživatelsky příjemnější pro programování vizuální podoby komponenty, jelikož zavádí koncepci plátna (*canvas*) a metodu *Paint()*.

Asi každého programátora zabývajícího se tvorbou komponent napadne zvolit za předka komponentu *Memo* nebo *RichEdit*. Bohužel se mi nepodařilo přijít na způsob, jak tyto komponenty rozšířit o možnost vykreslování malých obrázků (smajlíků), proto jsem tedy zvolil za předka třídu *TCustomControl*. S volbou této třídy je ale spojeno více práce s implementací, protože se musí naimplementovat reakce na stisky kláves, vykreslování textu na komponentu, posouvání textu, když je už textu tolik, že se nevejde do okna komponenty, implementace textového kurzoru atd. S tím vznikly některé potíže, které se mi nepodařilo zatím odstranit a popisují je v následující kapitole.

Psaní textu s touto komponentu probíhá očekávaným způsobem, komponenta reaguje na stisky kláves voláním vnitřních funkcí a vykresluje na sebe text. Znaky komponenta ukládá do statického pole, při překročení velikosti tohoto pole dojde k vyvolání události *OnFullEvent* a k smazání textu komponenty. Programátor má možnost v reakci na tuto událost zkopírovat text komponenty, popř. zareagovat jiným způsobem. Každý prvek pole pro ukládání znaků má typ záznamu, do kterého se ukládá, o jaký znak se jedná, dále pozice pravého horního bodu v pomyslném obdélníku, který znak resp. smajlík zabere při vykreslování, barvu a tučnost znaku a index do pole smajlíků. Vkládání smajlíků je řešeno přes *ImageList*, který se připojí ke komponentě, a index v záznamu o každém prvku je indexem do připojeného *ImageListu*. Pokud je tato hodnota rovna -1, jedná se o znak, jinak

je tento prvek smajlík. Ukládání pravého horního bodu (složka x, y) má výhodu v jednodušší implementaci pohybu kurzoru, jelikož je ke každému prvku známa jeho pozice, kdy na hodnoty x, y zobrazuje textový kurzor. Při změně obsahu komponenty je zavolána procedura, která ke každému prvku pole vypočítá hodnoty x a y a na konci zneplatní obsah komponenty, na což operační systém reaguje zasláním zprávy pro vykreslení a komponenta v události *Paint* (reakce na zprávu o překreslení) projde pole s prvky a vykreslí je. Pokud se celý text nevejde na komponentu, jsou v pravých rozích zobrazeny šipky, které signalizují, že se ještě nějaký text nachází „nad“ resp. „pod“ komponentou. Při pohybu po textu textovým kurzorem dochází případně k automatickému rolování textu. Označování textu do bloku je naimplementováno pouze pro myš. Komponenta si v události na stisknutí tlačítka myši dolů zaznamená začátek (index v poli prvků) a při uvolnění tlačítka myši konec (opět index v poli prvků) označeného textu a při vykreslování zvýrazní prvky pole, které se nacházejí mezi startovním a konečným indexem do pole prvků včetně. Takto označený text je možno přes interní *PopUpMenu* (vyvolá se stisknutím pravého tlačítka myši) např. zkopírovat do schránky. Přes toto menu je také možnost na pozici kurzoru vložit text ze schránky. Jelikož komponenta pro odřádkování používá pouze znak CR, dochází při kopírování/vkládání textu do/ze schránky ke konverzi řetězce. Při označeném textu včetně smajlíka do bloku a volbě kopírovat komponenta vrací text s náhradou smajlíka jeho příslušným řetězcovým kódem.

6.1.3 Problémy s implementací

Pokud má komponenta *focus* neboli má zaměření a je stisknuta klávesa na klávesnici, je vygenerována zpráva o této události a je zaslána objektu se zaměřením. V Delphi na tuto zprávu komponenta reaguje vyvoláním příslušné události. Např. při stisku klávesy na klávesnici je vyvolána událost *OnKeyDown*. V případě komponenty *IcqMemo* to až na kurzorové šipky funguje. Bohužel při stisku kurzorových šipek (nahoru, dolů, doleva a doprava) nedojde k vyvolání události *OnKeyDown*, ale dojde k přesunu *focusu* na další komponentu na formuláři. Toto chování se mi nepodařilo odstranit, a proto je potřeba, aby programátor používající tuto komponentu odchytil zprávy o stisknutí kurzorových šipek a volal public metodu komponenty *Kl_Sipka* s jedním parametrem, který reprezentuje virtuální kód šipky (např. šipka doleva má virtuální kód 37). Po zavolání této procedury komponenta zareaguje očekávaným způsobem.

Druhá vada této komponenty spočívá ve vytváření textového kurzoru. Podle literatury se v události *OnEntr* vytvoří textový kurzor a v události *OnExit* se textový kurzor odstraní. Bohužel pokud máme v aplikaci dva formuláře a na obou dvě komponenty, při kliknutí na první komponentu na formuláři jedna dostane tato komponenta focus a dojde k vyvolání události *OnEntr*. Při kliknutí na další komponentu na formuláři jedna je této další komponentě předán focus. U komponenty mající focus do této doby dojde k vyvolání události *OnExit* a u komponenty, která právě získala focus, je vyvolána událost *OnEntr*. Toto je očekávané chování a při implementaci, kdy v události *OnEntr*

vytvoříme kurzor a v události *OnExit* zrušíme kurzor, vše pracuje tak, jak má. Pokud jsme ale klikli na komponentu na prvním formuláři a ona dostala focus a dále klikneme na komponentu na jiném formuláři ve stejné aplikaci, nová komponenta dostane focus a vyvolá se s tím spojená událost *OnEntr*, ale u komponenty na formuláři jedna nedojde k vyvolání události *OnExit*, takže pokud opět klikneme na komponentu na formuláři jedna, nedojde k vyvolání události *OnEntr* a tedy k vytvoření textového kurzoru. Tento problém obchází komponenta následovně: v události *OnExit* je kurzor rušen, ale událost *OnEntr* není použita. Textový kurzor je vytvářen v události *OnClick*, která se vyvolává při kliknutí na komponentu myši. Bohužel tak někdy nedochází k dvojici volání *create* a *destroy* (vytvoření a zrušení) na textový kurzor, což samozřejmě není programátorsky nejčistší. Zatím se mi nepodařilo tento problém odstranit. I přes zmíněné chyby je komponenta použitelná a funguje vcelku dobře.

6.2 Komponenta IcqMemoZ

6.2.1 Požadavky

Komponenta bude zobrazovat komunikaci mezi dvěma uživateli. Bude umět označit text a zkopírovat ho do schránky. Na pravé straně bude zobrazen vertikální ScrollBox, kterým se bude posouvat text po komponentě, aby se uživatel mohl podívat na předešlou komunikaci nebo si případně postupně zobrazit příchozí zprávu, jestli bude natolik dlouhá.

Příchozí a odeslané zprávy budou zobrazeny odlišnou barvou (resp. ne zprávy, ale titulek zpráv), takže komponenta bude umožňovat měnit barvu a tučnost písma.

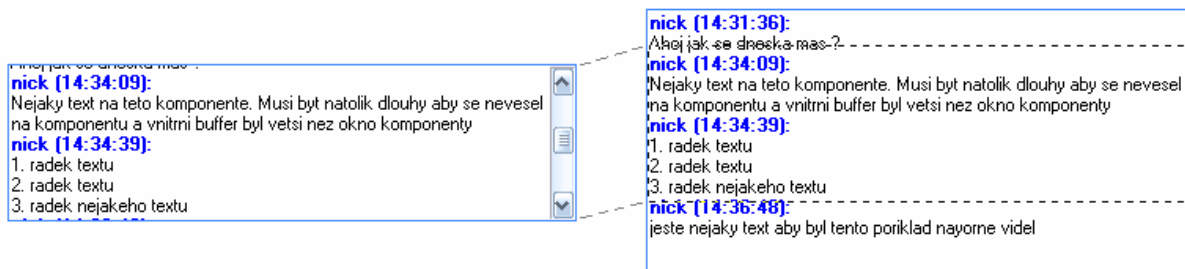
6.2.2 Implementace

Tato komponenta vychází z komponenty IcqMemo, je to vlastně její odlehčená verze, která slouží k zobrazování komunikace mezi dvěma uživateli. Neimplementuje textový kurzor.

Poskytuje metodu *Add* s parametrem řetězce, který má být zobrazen (přidán) na komponentě. Tato metoda automaticky odřádkuje a vloží text, pokud v textu najde řetězcový kód smajlíka, zobrazí ho na sobě místo tohoto kódu.

Pro znaky, které vykresluje, je použito jako u komponenty IcqMemo pole prvků. Po přidání textu do komponenty výše zmíněnou metodou *Add* je zavolána procedura, která veškerý platný text v poli vykreslí do vnitřního bufferu, který je široký jako komponenta a natolik vysoký, aby se na něj vešel všechny text, zneplatní obsah a v události *Paint* dochází už jenom k překreslení tohoto bufferu na komponentu. Posouvání textu je řešeno vertikálním scrollbarem umístěným u pravého okraje komponenty, posunutí tímto scrollbarem nastavuje y-ovou hodnotu při překreslení bufferu na

komponentu a na komponentě tak je zobrazen jen text, který odpovídá posunutí scrollbar, viz Obrázek 1.



Obrázek 1: Nalevo se nachází okno komponenty, napravo vnitřní buffer komponenty

Ostatní funkčnost této komponenty, jako např. použití jenom znaku CR pro odřádkování nebo označování textu do bloku a jeho kopírování do schránky, je řešeno podobně jako u komponenty IcqMemo.

7 Závěr

Cílem práce byla implementace klienta pro ICQ. Klient byl naprogramován v prostředí Delphi. Zastává základní funkčnost a přidává několik nadstandardních funkcí, které chybí v originálním klientovi a na pár výjimek i v alternativních klientech. Všechny tyto vymoženosti jsou vyjmenované v kapitole 5.1.1 a podařilo se je naimplementovat. Klient se pokouší nastínit směr, kterým by se měli vydat současní klienti, a to takový, že by s novými verzemi klientů přibývaly především další užitečné funkce, které zpříjemňují používání těchto „kecálek“, a ne jenom vizuální efekty, které akorát zvyšují hardwarové nároky.

Práce poskytuje příklad návrhu komponent v Delphi pro psaní textu s vkládáním smajlíků a jejich implementaci.

Další vývoj klienta by se měl soustředit na opravu a optimalizaci komponent `IcqMemo` a `IcqMemoZ`, o jejichž vadách jsem se zmínil v páté kapitole. Také by bylo vhodné se zabývat ukládáním historie, jelikož soubory s historií mohou narůst do velikostí, kdy už jenom jejich načtení by mohlo být časově neúnosné a všechny funkce spojené s ukládáním či čtením těchto souborů by mohly být dosti pomalé. Některé funkce související s protokolem nebo třeba funkce pro import historie zpráv z originálního klienta by bylo dobré přesunout do dll knihoven – zjednodušil by se update programu. Největší výzvou do budoucna je však nastudování ICQ protokolu a potažmo komponenty `TICQClient`, její vylepšení a vydávání aktualizací jak při změně protokolu, tak pro nové funkce.

Literatura

- [1] Fritzler, A.: *AIM/Oscar Protocol Specification* [online]. Poslední modifikace: 8. dubna 2000 [cit. 1.5.2007]. Dostupný na WWW: <<http://www.oilcan.org/oscar/>>
- [2] Charvát, Pavel.: *Komunikační software (ve znamení ICQ) aneb "kecejme po netu". PCtuning* [online]. 14.2.2005 [cit. 1.5.2007]. Dostupný na WWW: <http://www.pctuning.cz/index.php?option=com_content&task=view&id=4374&Itemid=96&limit=1&limitstart=0>
- [3] Mařovka, Marek.: *Už Vás štve otesánek ICQ5.1 ? Zkuste QIP!. PCtuning* [online]. 15.11.2006 [cit. 1.5.2007]. Dostupný na WWW: <http://www.pctuning.cz/index.php?option=com_content&task=view&id=7792&Itemid=96&limit=1&limitstart=0>
- [4] *QIP – icq klient. JPtuning* [online]. 4.12.2005 [cit. 1.5.2007]. Dostupný na WWW: <<http://www.jptuning.net/clanky/qip-icq-klient-46.html>>
- [5] Kubeš, Radek.: *Zapomeňte na ICQ, vyzkoušejte konkurenci. technet* [online]. 25.3.2007 [cit. 1.5.2007]. Dostupný na WWW: <http://technet.idnes.cz/zapomente-na-icq-vyzkousejte-konkurenci-fpz-/software.asp?c=A070322_170911_software_vse>
- [6] *OSCAR protocol. Wikipedia* [online]. 8.3.2003 [cit. 1.5.2007]. Dostupný na WWW: <http://en.wikipedia.org/wiki/OSCAR_protocol>
- [7] *ICQ. Wikipedia* [online]. 1.5.2007 [cit. 1.5.2007]. Dostupný na WWW: <<http://cs.wikipedia.org/wiki/ICQ>>
- [8] *Instant messaging. Wikipedia* [online]. 5.4.2007 [cit. 1.5.2007]. Dostupný na WWW: <http://cs.wikipedia.org/wiki/Instant_messaging>
- [9] Teixeira S., Pacheco.: X.: *Borland Delphi průvodce vývojáře kniha II*. Brno, 1999. Přel. Z DELPHI 4 Developer's Guide. ISBN: 80-86097-28-5.
- [10] Teixeira S., Pacheco.: X.: *Borland Delphi průvodce vývojáře kniha III*. Brno, 1999. Přel. Z DELPHI 4 Developer's Guide. ISBN: 80-86097-35-8.
- [11] Teixeira S., Pacheco.: X.: *Borland Delphi průvodce vývojáře kniha II*. Brno, 1999. Přel. Z DELPHI 4 Developer's Guide. ISBN: 80-86097-36-6.

Seznam použitých zkratek a symbolů

AIM – AOL Instant Messenger

CLX - Component Library for Cross Platform

DLL - Dynamic-link library

FTP - File Transfer Protocol

GNU – GNU's Not Unix

ICQ - protokol pro instant messaging a zároveň komunikační software

IM – instant messaging

OSCAR – Open System for CommunicAtion in Realtime

PC – personal computer

RAD - Rapid application development

RAM – Random access memory

RSS - Rich Site Summary

RTF - Rich Text Format

SDK – software development kit

UIN – Unified Identification Number

VCL - Visual Component Library

XML - The Extensible Markup Language

XMPP - The Extensible Messaging and Presence Protocol

Seznam příloh

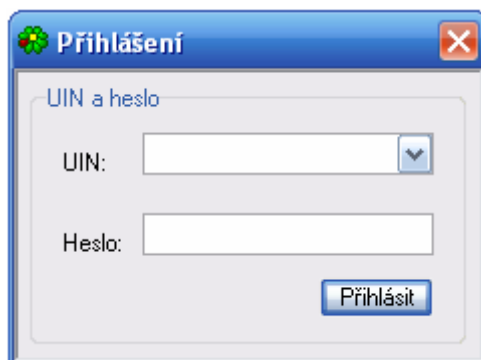
Příloha 1. Návod k použití

Příloha 2. Datový nosič CD s programem ICQ klient

Návod použití ICQ klienta

Formulář pro přihlášení

Po spuštění je spolu s hlavním oknem aplikace zobrazen přihlašovací formulář (viz. Obrázek 2), ve kterém je nutno vyplnit UIN (popřípadě vybrat z nabídky pokud se nejedná o první přihlášení) a heslo.

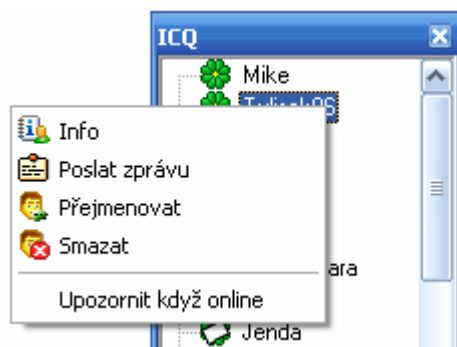


Obrázek 2: Formulář pro přihlášení

Pokud byly údaje správně vyplněny, klient načte seznam kontaktů a podle nastavení zobrazí kontakty buď se zařazením do skupin, nebo bez nich. Kliknutí na kontakt pravým tlačítkem myši vyvolá nabídku akcí souvisejících s kontaktem (viz. Obrázek 2).

Místní nabídka akcí pro práci s kontaktem

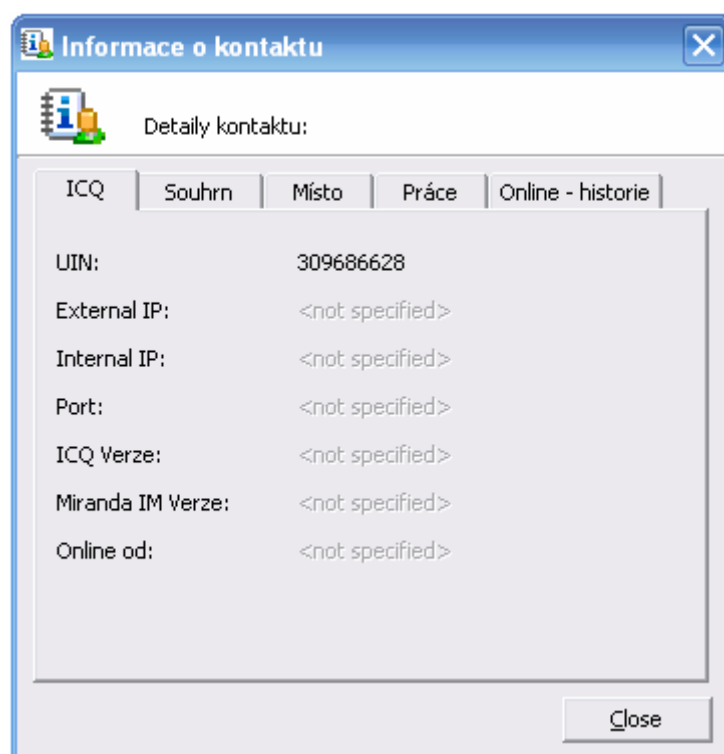
- **Info** – bude zobrazen formulář s informacemi o kontaktu
- **Poslat zprávu** – bude zobrazen formulář pro psaní zpráv
- **Přejmenovat** – zobrazení formuláře pro přejmenování kontaktu, změna se projeví také na serveru.
- **Smazat** – dojde k smazání kontaktu ze seznamu kontaktů
- **Upozornit, když online** – klient upozorní na status online u tohoto kontaktu



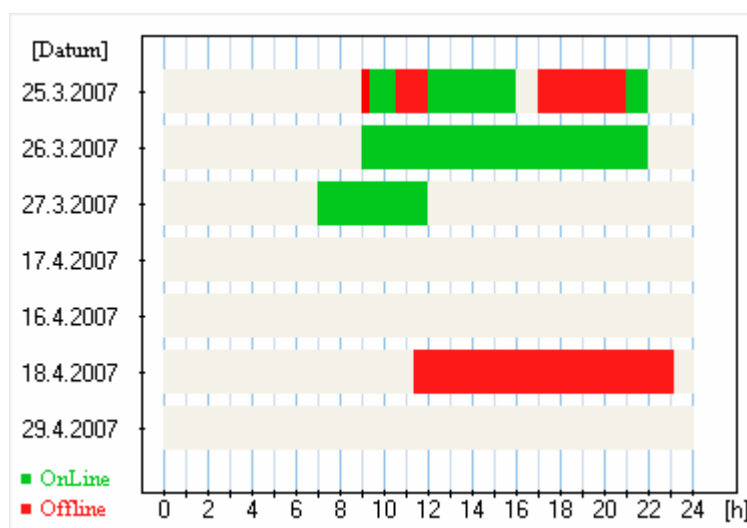
Obrázek 3: Hlavní okno aplikace se seznamem akcí pro kontakt

Formulář s informacemi o kontaktu

Na formuláři s informacemi o kontaktu (viz. Obrázek 3) jsou přehledně zobrazeny informace o kontaktu, jako je jeho jméno, příjmení, UIN, nick, datum narození atd. Tento formulář také slouží k zobrazení grafu, který reprezentuje historii připojení kontaktu k síti ICQ. Data pro tento graf (viz Obrázek 4) jsou sbírána jen při připojení klientovi.



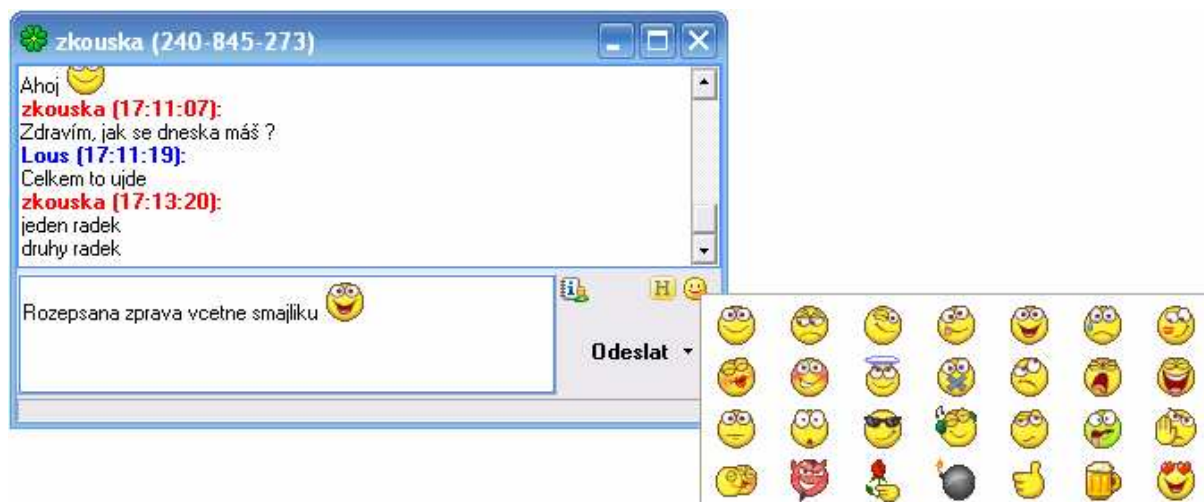
Obrázek 4: Informace o kontaktu



Obrázek 5: Graf připojení kontaktu

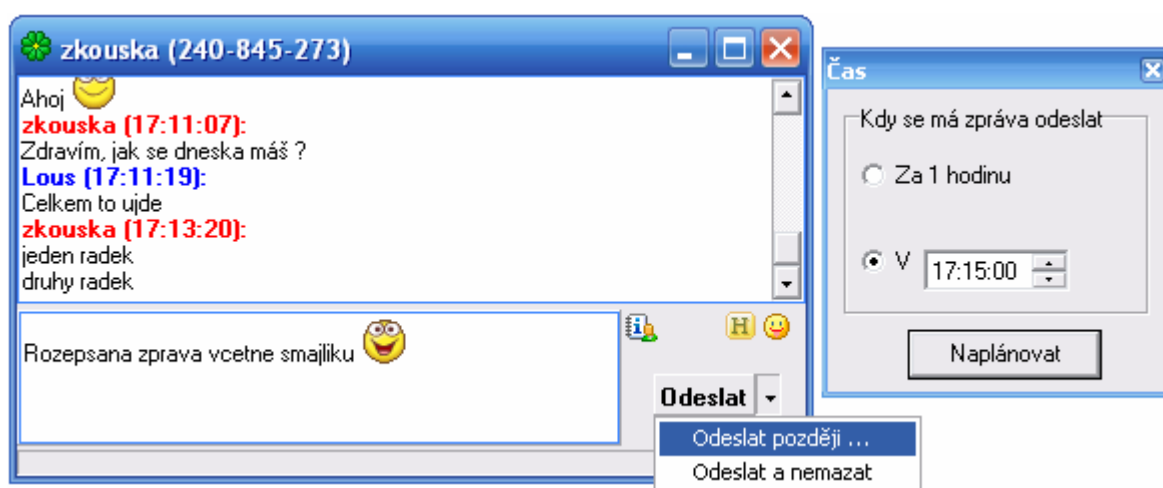
Formulář pro psaní a příjem zpráv

Volba poslat zprávu, stejně jako doubleklik, na kontaktu vyvolá formulář pro psaní a příjem zpráv (viz Obrázek 5). Kliknutí na ikonu smajlíka vyvolá formulář pro vložení smajlíka (jak je vidět na Obrázku 6).



Obrázek 6: Formulář pro komunikaci s kontaktem včetně formuláře se smajlíky

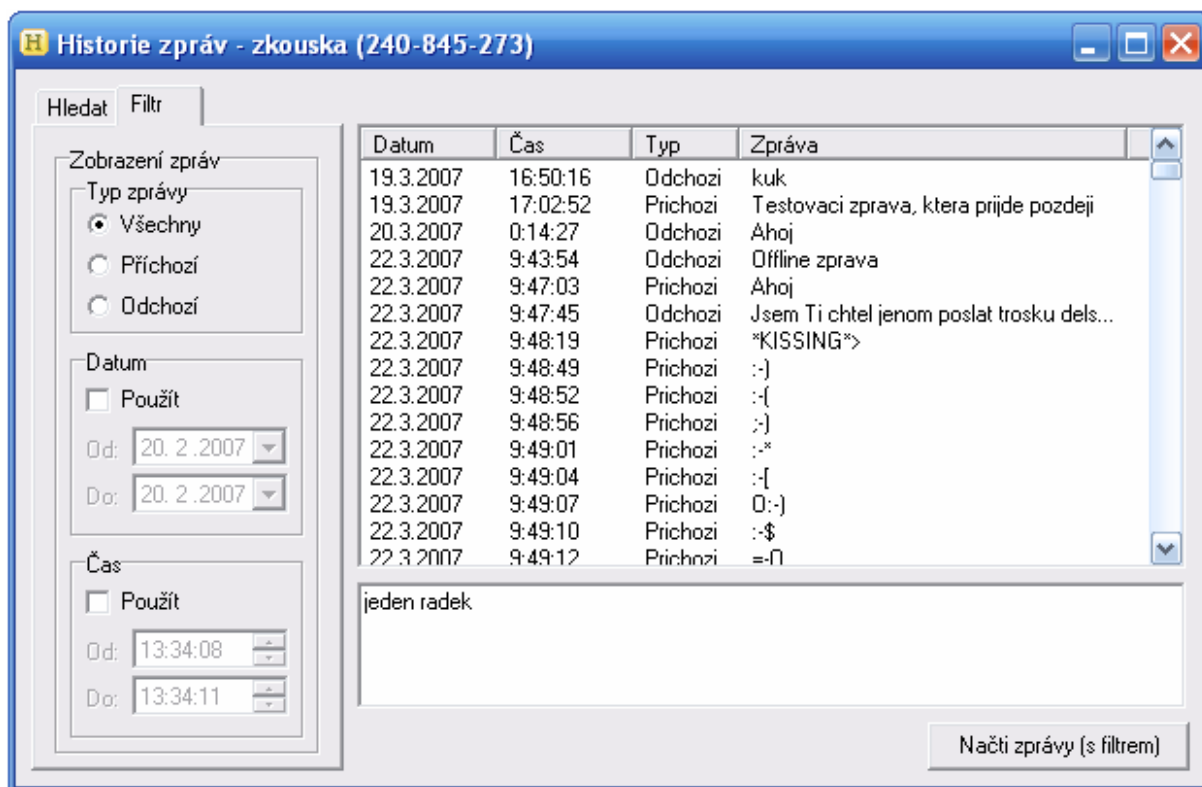
Uživatel má také možnost napsanou zprávu odeslat později, k čemuž slouží volba „Odeslat později“ na tlačítku s názvem „odeslat“ na formuláři pro psaní a příjem zpráv.



Obrázek 7: Formulář pro psaní a příjem zpráv, včetně ukázky pro odeslání zprávy později

Formulář s historií přijatých a odeslaných zpráv

Kliknutím na ikonu historie se zobrazí formulář s historií (viz Obrázek 6). Uživatel může v této historii vyhledávat a na zobrazené zprávy aplikovat filtr.



Obrázek 8: Formulář s historií

Hlavní menu aplikace

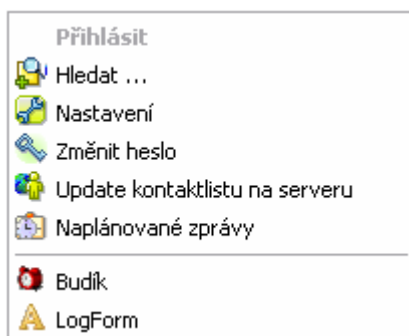
K vyvolání hlavního menu dojde kliknutím na tlačítko menu v hlavním okně (viz Obrázek 9).



Obrázek 9: Tlačítko menu na hlavním formuláři

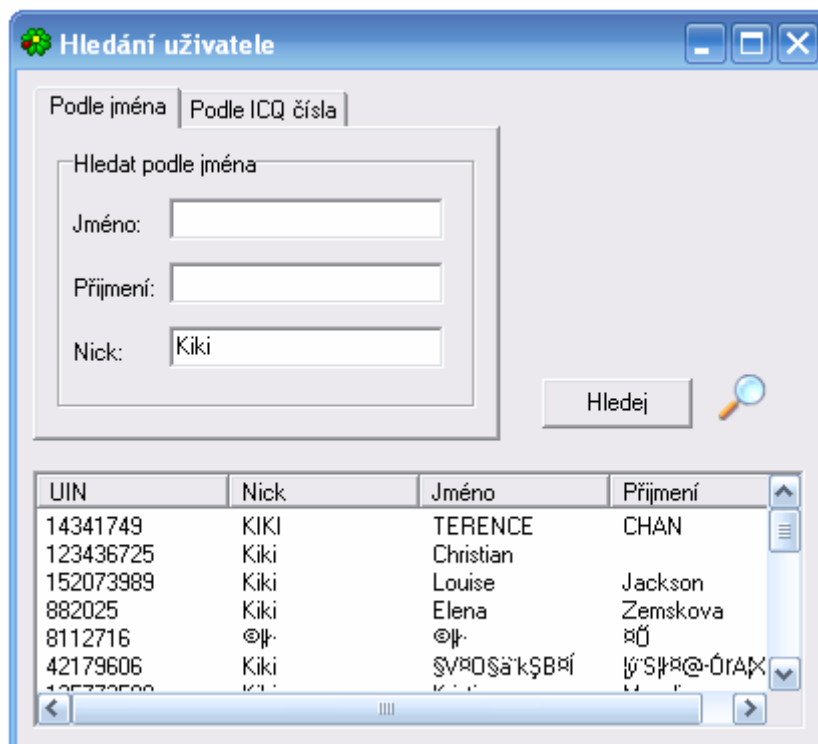
Hlavní menu aplikace (viz. Obrázek 10) nabízí tyto funkce:

- **Hledat** – volba zobrazí formulář pro hledání kontaktů
- **Nastavení** – zobrazí formulář s nastavením
- **Změnit heslo** – zobrazí formulář pro změnu hesla
- **Update kontaktlistu na serveru** – volba uloží všechny nově přidané kontakty na server
- **Naplánované zprávy** – zobrazí formulář se seznamem naplánovaných zpráv
- **Budík** – zobrazí formulář pro ovládání budíku
- **LogForm** – zobrazí logovací formulář, do kterého aplikace zaznamenává důležité informace při běhu programu (pro uživatele většinou nepodstatné)
- **Přihlásit** – volba zobrazí přihlašovací formulář



Obrázek 10: Hlavní menu aplikace

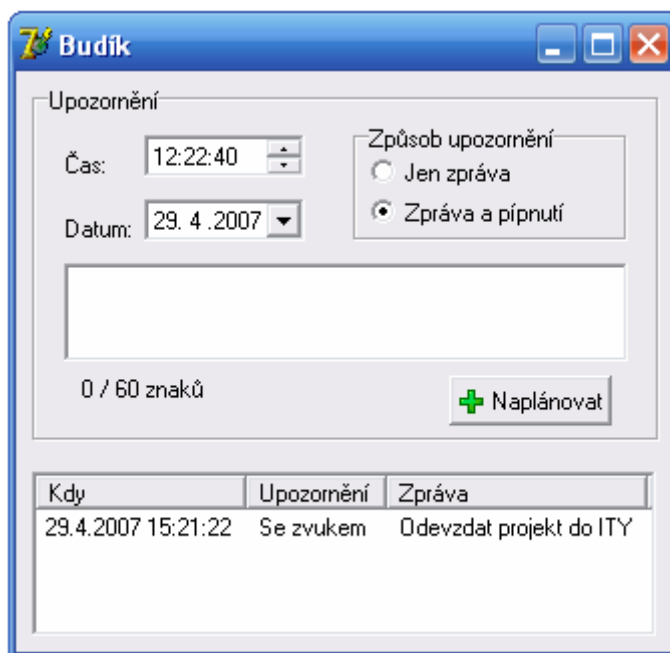
Formulář pro hledání kontaktů. Uživatel může vyhledávat podle jména, příjmení, nicku a nebo podle UIN.



Obrázek 11: Formulář pro hledání kontaktů

Formulář s budíkem

Formulář s budíkem (viz. Obrázek 12) umožňuje uživateli nastavit upomínku na stanovenou dobu a čas. Program tyto upomínky ukládá do souboru, takže nedochází ke smazání nastavených upomínek ani při vypnutí klienta.

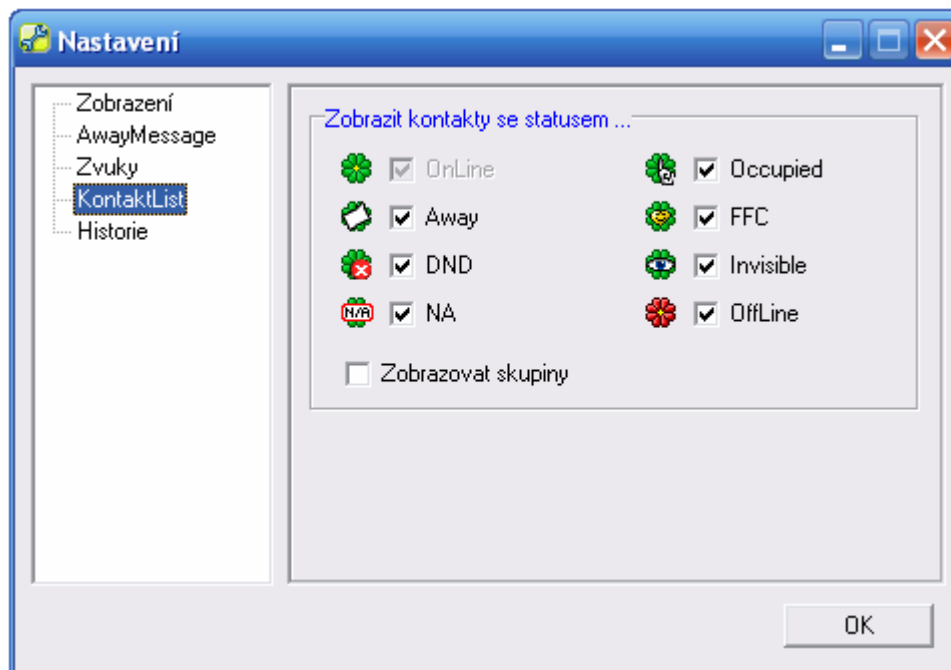


Kdy	Upozornění	Zpráva
29.4.2007 15:21:22	Se zvukem	Odevzdat projekt do ITY

Obrázek 12: Formulář s budíkem

Formulář s nastavením

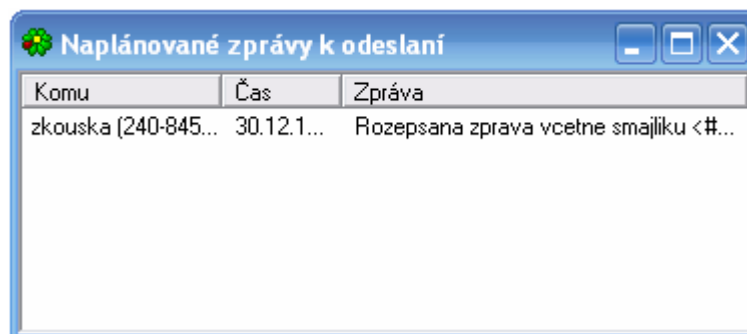
Volba nastavení zobrazí formulář s nastavením (viz. Obrázek 13). Uživatel si může zapnout popř. vypnout zvuk, nastavit zobrazení skupin, nastavit průhlednost hlavního okna a další věci. Přes tento formulář se také v možnostech historie importuje historie zpráv.



Obrázek 13: Formulář s nastavením

Formulář s naplánovanými zprávami

Tento formulář zobrazuje seznam naplánovaných zpráv (viz. Obrázek 14). Zde jde naplánovaná zpráva zrušit. Takto odeslaná zpráva je uložena v programu a ve stanovený čas odeslána.



Obrázek 14: Formulář s naplánovanými zprávami

Seznam obrázků

Obrázek 2: Formulář pro přihlášení	28
Obrázek 3: Hlavní okno aplikace se seznamem akcí pro kontakt	28
Obrázek 4: Informace o kontaktu	29
Obrázek 5: Graf připojení kontaktu	29
Obrázek 6: Formulář pro komunikaci s kontaktem, včetně formuláře se smajlíky	30
Obrázek 7: Formulář pro psaní a příjem zpráv, včetně ukázky pro odeslání zprávy později.	30
Obrázek 8: Formulář s historií	31
Obrázek 9: Tlačítko menu na hlavním formuláři	31
Obrázek 10: Hlavní menu aplikace	32
Obrázek 11: Formulář pro hledání kontaktů	32
Obrázek 12: Formulář s budíkem	33
Obrázek 13: Formulář s nastavením	34
Obrázek 14: Formulář s naplánovanými zprávami	34