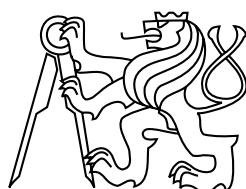


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Diplomová práce

Software pro správu mobilního telefonu



2004

Michal Čihař

Poděkování:

Úvodem této diplomové práce bych rád poděkoval jejímu vedoucímu inženýru Janu Kubrovi za cenné rady a připomínky, které přispěly k jejímu vzniku.

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

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

V Praze dne

Podpis

Annotation

The aim of this thesis is to create software for managing data stored in mobile phone. Currently it is possible to keep much data in phones (like contacts, todos and calendar events), but there is no application for Linux operating system, which would allow to access these data by identical way for many different phones. This gap should be filled in by results of thesis.

Anotace

Cílem této práce je vytvořit program pro správu dat uložených v mobilním telefonu. V současné době je možné do telefonů ukládat mnoho údajů, například kontakty, úkoly a události do kalendáře, a neexistuje aplikace, která by fungovala na operačním systému Linux a podporovala velké množství telefonů. Tato mezera by měla být zaplněna výsledkem této práce.

Zadání

Navrhněte a implementujte program pro správu mobilního telefonu. Při řešení se zaměřte na univerzálnost, podporu národních jazykových sad a možnosti konverze dat mezi různými mobilními telefony. Diskutujte možnosti synchronizace s jinými aplikacemi.

Obsah

1	Úvod	1
1.1	Vysvětlení některých zkratk	1
1.2	Přílohy	2
2	Existující projekty	3
2.1	Gammu	3
2.2	Gnokii	3
2.3	Gsmlib	4
2.4	Další projekty	4
2.5	Software pro Windows	4
2.5.1	Software dodávaný výrobcem	5
2.5.2	APDS	5
2.5.3	Oxygen Phone Manager	5
2.5.4	Handset Manager	5
2.5.5	Alcatel manažer	5
3	Popis Gammu	6
3.1	Struktura projektu	6
3.1.1	Modul zařízení	7
3.1.2	Modul protokolu	7
3.1.3	Modul telefonu	8
3.2	Funkce Gammu	9
3.2.1	Informace o telefonu	9
3.2.2	Nastavení telefonu	10
3.2.3	Kontakty a hovory	11
3.2.4	Zprávy	11
3.2.5	Úkoly	12

3.2.6	Kalendář	12
3.2.7	Další data v telefonu	13
3.2.8	Soubory v telefonu	13
3.2.9	Funkce telefonu	14
3.2.10	Notifikace událostí	14
3.2.11	Formát dat	15
3.2.12	Další poskytované funkce	15
3.2.13	Ošetřování chyb	16
3.3	Moje práce na projektu	17
3.3.1	Modul pro telefony Alcatel	17
3.3.2	Modul pro AT příkazy	19
4	Grafické uživatelské rozhraní	20
4.1	Grafická knihovna	20
4.1.1	GTK	20
4.1.2	Qt	21
4.1.3	wxWindows	21
4.2	Programovací jazyk	21
5	python-gammu	22
5.1	Reprezentace dat	22
5.2	Struktura modulu	23
5.3	Výjimky	23
5.4	Funkce Gammu	23
6	Wammu	25
6.1	Vzhled a jeho implementace	25
6.1.1	Zobrazení objektů	25
6.1.2	Editace kontaktů, úkolů a událostí kalendáře	26
6.1.3	Editace zpráv	27
6.1.4	Zobrazení zpráv	28
6.2	Komunikace s telefonem	29
6.3	Vyhledávání telefonu	29
6.4	Export a import dat	30
6.5	Lokalizace	30
6.6	Požadavky	31

7 Synchronizace	32
7.1 Problémy	32
7.2 Existující projekty	32
7.2.1 KnokiiSync	33
7.2.2 Multisync	33
7.3 Shrnutí	33
8 Zhodnocení	34
Literatura	35
Rejstřík	37

Kapitola 1

Úvod

Tato práce byla motivována nedostupností kvalitního softwarového vybavení pro správu dat v mobilních telefonech, které postupně umožňují organizovat stále více informací, pro operační systém *Linux*. Vzniklá aplikace by měla být schopná komunikovat s co možná největším množstvím telefonů a umožňovat snadnou editaci uložených dat.

Stávající projekty jsou většinou zaměřené jen na jeden druh případně výrobce telefonů nebo jsou natolik nedokončené, že je jejich používání téměř nemožné. Kromě toho žádná aplikace neumožňuje snadnou migraci dat při změně telefonu, což je funkcionality, která mi citelně chyběla.

Kromě operačního systému *Linux*, by aplikace rovněž měla fungovat i na dalších systémech, za maximálního využití na platformě nezávislých prvků.

1.1 Vysvětlení některých zkratk

V textu se často vyskytují některé zkratky a pojmy, které nemusí být známy, proto zde vysvětlím jejich význam.

AT příkazy AT příkazy jsou standardní způsob komunikace s modemy definované původně ITU. Pro potřeby mobilních telefonů byl jejich rozsah značně rozšířen dalšími normami vydanými *ETSI* (*GTS*).

Bluetooth Standard pro radiové propojení různých periférií (mobilní telefony, digitální fotoaparáty, tiskárny atd.) na krátký dosah – <http://www.bluetooth.com>.

GSM *Global System for Mobile Communications* – Globální systém pro mobilní komunikaci – <http://www.gsmworld.com/>.

GTS *GSM Technical Specification* – Technické specifikace *GSM* sítí vydávané *ETSI*. Kromě technických parametrů provozu sítě definují i *AT* příkazy, kterými by mělo být možné se zařízením komunikovat.

EMS *Enhanced Messaging Service* – Rozšíření možností *SMS* o formátování textu a o možnost vložit obrázky a melodie.

ETSI *European Telecommunications Standards Institute* – Evropský úřad pro standardy v telekomunikacích – <http://www.etsi.org>.

IrDA *Infrared Data Association* – nezisková organizace zaštiťující standardy pro infračervené propojení různých zařízení – <http://www.irda.org>.

OBEX *Object Exchange* – Protokol pro výměnu dat, definovaný ve specifikaci Bluetooth – <http://www.bluetooth.com>.

PDU *Protocol Data Unit* – jednotka dat přenášená po síti. V oblasti *GSM* sítí je tento název často používán i pro formát *SMS* posílaných po síti.

Smart Messaging Podobně jako *EMS* se jedná o rozšíření *SMS*, na rozdíl od *EMS*, se však nejedná o standard akceptovaný všemi výrobci, původní návrh vytvořila firma *Nokia*.

SMS *Short Message Service* – Služba umožňující posílat krátké zprávy prostřednictvím sítě *GSM*.

ITU *International Telecommunications Union* – Mezinárodní telekomunikační unie – <http://www.itu.ch/>.

WAP *Wireless Application Protocol* – Síťový protokol používaný pro bezdrátovou komunikaci. Je standardizován *WAP* fórem – <http://www.wapforum.org>.

1.2 Přílohy

Nedílnou součástí celé práce je též přiložené CD-ROM obsahující kompletní zdrojový text programů, jejich spustitelnou verzi pro některé distribuce *Linuxu* a vygenerovanou dokumentaci k *API Gammu*.

CD-ROM obsahuje také Live distribuci Linuxu – *Damn Small Linux*, kterou je možné naboootovat a z ní spustit *Wammu*. Tato distribuce má však z důvodu minimalizace obsazeného prostoru velmi omezenou podporu locales, takže práce s národními znaky může způsobit problémy.

Další přílohou, tentokrát tištěnou, je changelog projektu *Gammu*, ze kterého je patrný vývoj projektu a moje příspěvky do něj.

Kapitola 2

Existující projekty

Protože pochopitelně nejsem první, kdo chce napsat takovouto aplikaci, předcházelo započítí práce na tomto projektu prozkoumání existujících projektů. Většina z nich se zaměřuje jen na jeden model, případně jednoho výrobce telefonů a není možné je snadno rozšířit pro podporu dalších modelů. To je často způsobeno tím, že vývojář píše jen pro svůj vlastní telefon, mimo jiné i proto, že nemůže testovat program s více telefony. Protože ani já nemám přístup k více telefonům, rozhodl jsem se přispět do již existujícího projektu přidáním podpory pro telefony značky *Alcatel* a dále vytvořením dalších navazujících aplikací. Dále popíši projekty, které přicházely v úvahu a jejich výhody či nevýhody.

2.1 Gammu

Projekt *Gammu* (<http://mwiacek.com/gsm/gammu/gammu.html>) vznikl odloučením několika vývojářů od *Gnokii*, kvůli nespokojenosti s rychlostí vývoje a celkovým návrhem kódu. Jedním z motivů byla také kompletní práce s texty v unikódu, čímž se zredukuje problémy s národními znaky.

Poslední uvedená věc byla jedna z rozhodujících, proč jsem se nakonec rozhodl rozšířit možnosti tohoto projektu. Podrobnosti o tomto projektu jsou uvedeny v kapitole 3 na straně 6.

2.2 Gnokii

Gnokii (<http://gnokii.org>) je knihovna a program pro komunikaci s telefony *Nokia* a některými dalšími telefony pomocí *AT* příkazů. Jedná se bezpochyby o nejznámější projekt v této oblasti, ale svým značným zaměřením na telefony *Nokia*, znesnadňuje rozšíření o jiné telefony.

Oproti *Gammu* má *Gnokii* poněkud jinak navržen automat, do kterého jsou jednotlivé moduly zasazeny – *Gammu* používá strukturu, ve které jsou přímo ukazatele na

dané funkce, kdežto *Gnokii* má operace definované jako výčtový typ a poté každý modul definuje seznam podporovaných operací. Pro větší množství nenadefinovaných funkcí je výhodnější toto řešení, v opačném případě řešení použité v *Gammu*. Až na toto a rozdíly zmíněné u popisu *Gammu*, jsou oba projekty velmi podobné.

2.3 Gsmlib

Gsmlib (<http://www.pxh.de/fs/gsmlib/>) je knihovna umožňující komunikovat s telefony pomocí *AT* příkazů. Je implementována podle *ETSI* specifikací a umožňuje vytvářet rozšíření pro další telefony, bohužel je počítáno jen s přidáváním dalších *AT* příkazů a nikoliv s přidáváním jiného způsobu komunikace.

Velkou výhodou tohoto projektu je značné využívání C++, například šablon a výjimek.

K této knihovně také existuje velmi jednoduché grafické uživatelské rozhraní, které umožňuje práci s kontakty a textovými zprávami.

2.4 Další projekty

Kromě výše uvedených existuje ještě několik projektů, které se buďto ani nikdy nesnažily o rozšiřitelnost, nebo jejich vývoj byl ukončen po vydání několika prvních verzí.

KAlcatel (<http://www.cihar.com/alcatel>) – program pouze pro jeden konkrétní model telefonu. Jedná se o projekt, který jsem vytvořil, ale po několika verzích jsem vývoj zastavil, protože bylo zřejmé, že nedomulární návrh je velkou nevýhodou.

Kandy (<http://kandy.kde.org/>) – jednoduchý program umožňující zadávat *AT* příkazy, je součástí desktopového prostředí *KDE*. Původní snahou bylo vyvinout podstatně komplexnější aplikaci, bohužel zůstalo jen u návrhu.

KSiemens (<http://sourceforge.net/projects/ksiemens>) – zaměřeno na telefony *Siemens*, vývoj zastaven ve velmi rané fázi.

2.5 Software pro Windows

Protože vytvořený software má být funkční mimo jiné i na platformě Windows, zajímal jsem se také o software dostupný pro tuto platformu. Tento seznam rozhodně není kompletní, zmiňuji zde jen programy, které mě zaujaly a se kterými jsem se setkal.

2.5.1 Software dodávaný výrobcem

Většina výrobců telefonů dodává různě kvalitní software již přímo k telefonu. Nevýhoda těchto programů je obvykle jejich omezená funkčnost a pak také to, že pokud se uživatel v budoucnu rozhodne změnit telefon, je nucen používat a zvykat si i na nový software.

2.5.2 APDS

All Phone Data Suite (<http://www.p-emtech.cz>) je český produkt, podporující značné množství telefonů, buhužel však umožňuje pracovat pouze s textovými zprávami a kontakty.

2.5.3 Oxygen Phone Manager

Oxygen Phone Manager (<http://www.oxygensoftware.com/>) je program určený pro mnoho telefonů firmy Nokia. Umožňuje komplexní správu všech dat v mobilním telefonu.

2.5.4 Handset Manager

Handset Manager (<http://www.mobileaction.com>) je určen především pro správu multimediálních dat v telefonech připojujících se přes infračervený port a používající ke komunikaci protokol *OBEX*.

2.5.5 Alcatel manažer

Alcatel manažer (<http://ace.ic.cz>) je program postavený na mojí knihovně pro komunikaci s telefony Alcatel (stejná jako je použita v projektu *KAlcatel* a umí tudíž spolupracovat jen s některými telefony firmy *Alcatel*.

Kapitola 3

Popis Gammu

Gammu vzniklo původně z projektu *Gnokii*, po neshodách o dalším směřování vývoje. Oproti *Gnokii* má kratší vývojový cyklus (verze vycházejí častěji), veškeré texty jsou zpracovávány v unicode, je podporováno větší množství telefonů a má poněkud jinou vnitřní strukturu.

Hlavním vývojářem a iniciátorem projektu je Marcin Wiacek. Moje příspěvky do projektu jsou podrobněji rozebrány v kapitole 3.3 na straně 17.

Následující část popisuje vnitřní strukturu *Gammu* a některé důležitější funkce. Na příloženém CD je kompletní vygenerovaná dokumentace, která z důvodu rozsahu není vytištěna.

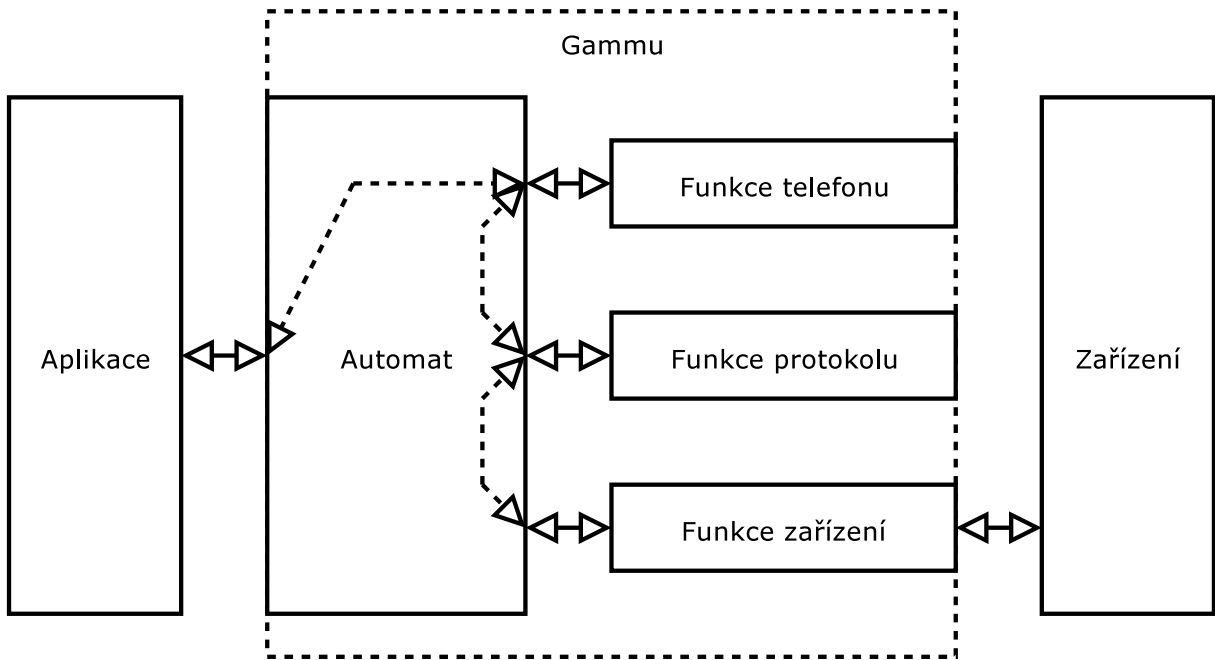
3.1 Struktura projektu

Hlavní část projektu tvoří automat, který definuje rozhraní pro ostatní moduly. Tyto moduly se dají rozdělit do tří částí – funkce zařízení, protokolu a telefonu. Komunikace probíhá přes definované rozhraní, takže vyšší vrstvy se nemusejí zajímat o nižší, pokud to není vynuceno jejich provázaností ve skutečnosti. Vnitřní struktura a její interakce s aplikací a zařízením jsou naznačeny na obrázku 3.1.

Jednotlivým částem se ještě budu věnovat podrobněji dále. Kromě toho ještě do komunikace mohou zasahovat uživatelem (respektive aplikací) definované funkce, které umožňují jednak reagovat na asynchronní události způsobené například přijetím zprávy, hovorem a podobě, ale také případně umožňuje v aplikační vrstvě přímo komunikovat s telefonem, například prostřednictvím *AT* příkazů.

Další částí *Gammu* je část nazvaná služby. Jedná se o funkce, které zajišťují práci s různými typy dat (např. dekodování *SMS* zpráv, konverze melodií a zálohování).

Informace o implementovaných modulech a funkcích mohou být zastaralé, protože program se neustále vyvíjí, popisována je verze 0.92.0.



Obrázek 3.1: Struktura Gammu

3.1.1 Modul zařízení

Modul zařízení umožňuje nízkourovňovou komunikaci prostřednictvím nějakého zařízení. Umí jen zapsat a přečíst posloupnost znaků a nastavit parametry zařízení. V současné době jsou implementovány moduly pro sériový port, *IrDA* a *Bluetooth*, všechny jak pro *Microsoft Windows*, tak pro *Linux*.

3.1.2 Modul protokolu

Tento modul má na starosti přijímání a odesílání zpráv. Odesílání se obvykle sestává z doplnění hlaviček a kontrolních součtů, přijímání pak v kontrole těchto dodatečných dat a jejich odstranění pro vyšší vrstvu.

Implementovány jsou tyto protokoly:

AT je standardní komunikace s telefony a modemy definovaná normami *ETSI* a *ITU*

OBEX je standardizovaný protokol pro přenos souborů, používá se především ve spojení s *IrDA* nebo *Bluetooth*, ale někteří výrobci jeho modifikaci používají i na sériovém připojení

mrouter je používán zařízeními používajícími operační systém *Symbian*

FBUS2 je proprietární protokol společnosti *Nokia*

MBUS2 je proprietární protokol společnosti *Nokia*

Phonet je proprietární protokol společnosti *Nokia*, používaný pro *IrDA* připojení

alcabus je proprietární protokol společnosti *Alcatel*

3.1.3 Modul telefonu

Na této úrovni jsou již nadefinovány funkce zajímavé pro uživatele, tedy vlastní práce s telefonem. Každý modul obsahuje jen nějakou část funkci, protože většina telefonů nepodporuje všechny funkce, které Gammu umožňuje používat.

Telefonů je podporováno mnoho: většina telefonů *Nokia*, některé typy *Alcatelů* a telefony podporující *AT* příkazy nebo přenos pomocí protokolu *OBEX*. Popis jednotlivých modulů je uveden dále.

Alcatel

Tento modul implementuje komunikaci s novějšími telefony *Alcatel* (konkrétně se jedná o modely BE5, BF5 a BH4). Tyto telefony komunikují pomocí *AT* příkazů a speciálním *AT* příkazem se dají přepnout do jiného komunikačního režimu, ve kterém je možné přistupovat k datům (telefonní seznam, kalendář a úkoly) uloženým v telefonu. Modul proto pro mnoho funkcí používá funkce definované pro *AT* telefony (modul je popsán dále).

AT

Komunikaci pomocí standardních *AT* příkazů zajišťuje tento modul. Kromě příkazů definovaných v normách *ETSI*, podporuje i rozšíření některých výrobců, především *Siemens*. Protože mnoho výrobců nedodržuje normy přesně a normy jsou v některých případech nejednoznačné (například není definováno pořadí bajtů v případě používání unicode nebo zda telefonní čísla mají být vypisována v unicode), je potřeba zde ošetřovat mnoho nestandardních situací. V současné době by se Gammu mělo vypořádat s většinou těchto situací a tudíž naprostá většina telefonů umějících *AT* příkazy je podporována.

Nokia

Toto je nejstarší modul a také obsahuje nevíce funkcí. Podporována je celá řada telefonů (jejich seznam je uveden v souboru `readme.txt`).

OBEX

Umožňuje pracovat s telefony, které obsah paměti zpřístupňují pomocí protokolu *OBEX*. Pomocí tohoto modulu je tedy možné jen pracovat se soubory a složkami v telefonu.

Symbian

Tento modul v současné době neobsahuje ještě žádnou funkcionalitu, ale v budoucnosti by měl podporovat zařízení postavená na operačním systému *Symbian*.

3.2 Funkce Gammu

Základní funkce komunikující s telefonem se dají rozdělit do několika částí, důležitější funkce jsou uvedeny v následujících kapitolách. Kompletní přehled, včetně popisu parametrů, je v dokumentaci projektu.

3.2.1 Informace o telefonu

Funkce umožňující získat informace o telefonu, jeho stavu a o síti, do které je přihlášen.

GetManufacturer Výrobce telefonu

GetModel Model telefonu

GetFirmware Firware telefonu

GetIMEI IMEI případně sériové číslo telefonu

GetOriginalIMEI Původní IMEI telefonu

GetNetworkInfo Informace o síti

GetManufactureMonth Měsíc výroby

GetProductCode Kód produktu

GetHardware Informace o hardware

GetPPM Informace o Post Programmable Memory (dostupné u telefonů *Nokia*. Tato paměť obsahuje například informace o nainstalovaných jazycích)

GetSecurityStatus Zjistí jestli telefon očekává zadání PIN/PUK/... kódu

GetDisplayStatus Stav displeje

GetSIMIMSI IMSI (*International Mobile Subscriber Identify*) SIM karty

GetBatteryCharge Nabití baterie a informace o zdroje napájení

GetSignalQuality Kvalita signálu

3.2.2 Nastavení telefonu

Změna a přečtení různých nastavení telefonu. Kromě základních věcí jako datum a čas, je u některých telefonů přístupné mnoho dalších voleb.

GetDateTime Přečte hodiny

SetDateTime Nastaví hodiny

GetAlarm Přečte budík

SetAlarm Nastaví budík

GetSMSC Přečte středisko SMS zpráv

SetSMSC Nastaví středisko SMS zpráv

GetWAPSettings Přečte nastavení WAPu

SetWAPSettings Změní nastavení WAPu

GetProfile Přečte profil voleb

SetProfile Nastaví profil voleb

GetSpeedDial Přečte rychlé vytáčení

SetSpeedDial Nastaví rychlé vytáčení

ResetPhoneSettings Obnoví původní nastavení telefonu

SetAutoNetworkLogin Zapne automatické přihlašování do sítě

GetMMSSettings Přečte nastavení MMS

SetMMSSettings Změní nastavení MMS

GetCallDivert Přečte přesměrování hovorů

SetCallDivert Nastaví přesměrování hovorů

CancelAllDiverts Zruší všechna přesměrování

GetGPRSAccessPoint Zjistí přístupový bod GPRS

SetGPRSAccessPoint Nastaví přístupový bod GPRS

GetLocale Přečte nastavení jazyku

SetLocale Změní nastavení jazyku

GetCalendarSettings Přečte nastavení kalendáře

SetCalendarSettings Změní nastavení kalendáře

3.2.3 Kontakty a hovory

Protože kontakty a hovory jsou z pohledu telefonu řešeny stejně, tak k nim i *Gammu* přistupuje stejně a souhrně jsou nazývány **Memory**. Protože český překlad mi nepřipadá ekvivaletní, mluvím dále o kontaktech, ale myslím tím oba dva druhy položek.

AddMemory Přidání kontaktu

SetMemory Změna kontaktu

DeleteMemory Vymazání kontaktu

DeleteAllMemory Vymazání všech kontaktů z dané paměti

GetMemory Přečtení jednoho kontaktu

GetNextMemory Přečtení dalšího kontaktu, tato funkce je optimalizována pro čtení celého seznamu

GetMemoryStatus Informace o stavu paměti – počet uložených kontaktů

AddCategory Přidání kategorie kontaktů

GetCategory Přečtení názvu kategorie kontaktů

GetCategoryStatus Zjištění počtu kategorií kontaktů

3.2.4 Zprávy

Úmyslně zde není použit název textové zprávy, protože podporováno je i mnoho dalšího obsahu, který může být ve zprávách posílán – například *EMS* a *Smart Messaging*.

AddSMS Přidání zprávy

SetSMS Změna zprávy

DeleteSMS Vymazání zprávy

DeleteAllSMS Vymazání všech zpráv z dané složky

GetSMS Přečtení jedné zprávy

GetNextSMS Přečtení další zprávy, tato funkce je optimalizována pro čtení všech zpráv

GetSMSStatus Informace o počtu zpráv

AddSMSFolder Přidání složky na zprávy

GetSMSFolders Přečtení složek na zprávy

DeleteSMSFolder Vymazání složky na zprávy

3.2.5 Úkoly

Práce s úkoly, jejich čtení, mazání a vytváření. Úkoly lze rozdělit u některých telefonů do kategorií, takže sem patří i funkce pro práci s nimi.

AddToDo Přidání úkolu

SetToDo Změna úkolu

DeleteToDo Vymazání úkolu

DeleteAllToDo Vymazání všech úkolů

GetToDo Přečtení jednoho úkolu

GetNextToDo Přečtení dalšího úkolu, tato funkce je optimalizována pro čtení všech úkolů

GetToDoStatus Informace o počtu úkolů

AddCategory Přidání kategorie úkolů

GetCategory Přečtení názvu kategorie úkolů

GetCategoryStatus Zjištění počtu kategorií úkolů

3.2.6 Kalendář

Práce s úkoly, jejich čtení, mazání a vytváření.

AddCalendar Přidání události

SetCalendar Změna události

DeleteCalendar Vymazání události

DeleteAllCalendar Vymazání všech událostí

GetCalendar Přečtení jedné události

GetNextCalendar Přečtení další události, tato funkce je optimalizována pro čtení celého kalendáře

GetCalendarStatus Informace o počtu událostí

3.2.7 Další data v telefonu

Mnoho telefonů zpřístupňuje i další data – například melodie, záložky pro *WAP*, obrázky (loga) a předvolby rádia.

GetRingtonesInfo Informace o vyzváněních

DeleteUserRingtones Vymazání uživatelem definovaných vyzvánění

GetRingtone Načtení vyzvánění

GetWAPBookmark Načtení záložky pro WAP

SetWAPBookmark Nastavení záložky pro WAP

GetBitmap Načtení obrázku

SetBitmap Nastavení obrázku

GetFMStation Přečtení předvolby FM rádia

SetFMStation Nastavení předvolby FM rádia

ClearFMStations Vymazání předvoleb FM rádia

3.2.8 Soubory v telefonu

Do některých telefonů se dají ukládat soubory a pro práci s nimi se dají použít následující funkce:

GetFileSystemStatus Stav souborového systému

GetNextFileFolder Výpis složky

GetFilePart Získání (části) souboru

AddFilePart Zapsání (části) souboru

DeleteFile Smazání souboru

AddFolder Vytvoření složky

3.2.9 Funkce telefonu

V této části je především práce s hovory, ale i některé obecnější funkce pracující s telefonem.

DialVoice Vytočí číslo a zahájí hlasový hovor

AnswerCall Zvedne příchozí hovor

CancelCall Zruší příchozí hovor

PressKey Simuluje stisk klávesy

PlayTone Zahraje tón

EnterSecurityCode Zadání bezpečnostního kódu PIN/PUK/...

SendDTMF Tónová volba

HoldCall Přidržení hovoru

UnholdCall Zvednutí přidrženého hovoru

ConferenceCall Konferenční hovor

SplitCall Rozdělení hovoru

TransferCall Přenesení hovoru

SwitchCall Přepnutí jiného aktivního hovoru

Reset Reset telefonu

3.2.10 Notifikace událostí

Pokud chce být aplikace informována o událostech, které může vyvolat síť, jako například příchozí hovor, SMS zpráva a podobně, musí tyto notifikace aktivovat.

SetIncomingSMS Nastaví notifikaci příchozích zpráv

SetIncomingCB Nastaví notifikaci cell broadcast informací

SetIncomingCall Nastaví notifikaci příchozích hovorů

SetIncomingUSSD Nastaví notifikaci dalších příchozích služeb

3.2.11 Formát dat

Všechna data, která mohou obsahovat proměnný počet položek (kontakty, kalendář a úkoly) jsou ukládána v poli, kde je určen pro každou položku její typ a také uložena její hodnota. Tím je umožněno pracovat s různými daty od různých telefonů a převádět je mezi sebou.

3.2.12 Další poskytované funkce

Kromě vlastní komunikace s telefony Gammu ještě nabízí další funkce pro práci s daty přečtenými z telefonu. Nejvýraznější je práce s textovými zprávami a zálohování, které budou popsány samostatně. Kromě toho je ještě obsažena podpora pro:

- konfigurační soubory
- úpravu a konverzi obrázků (například loga)
- úpravu a konverzi melodií
- zjištění názvu sítě a země ve které se nachází
- konverzi interních dat pro jednotlivé položky (kontakty, kalendář a úkoly) na standardní formáty používané například v PIM aplikacích - vCard, vCalendar a vTodo

Zprávy

Zprávy jsou částí, která si zaslouží nejvíce pozornosti. Umožňuje totiž pracovat s mnoha typy zpráv i na telefonech, které s nimi samy pracovat neumí. To je umožněno jednak podporou spojování dlouhých zpráv a také podporou mnoha druhů zpráv:

- *EMS* – zprávy mohou obsahovat obrázky, melodie a formátování textu
- *Smart Messaging* (rozšíření telefonů *Nokia*) – profily, obrázky a melodie
- nastavení *MMS* a *WAP* (pro telefony *Sony Ericsson* a *Nokia*)
- vCard a vCalendar – standard pro posílání událostí a kontaktů
- *MMS* indikátor – obsahuje *URL* odkud má telefon stáhnout *MMS* zprávu
- šablony zpráv pro telefony *Alcatel*

Podporováno je jak dekodování těchto zpráv, tak jejich vytváření.

Melodie

Melodie je možné převádět mezi mnoha formáty podporovanými různými telefony, ale i exportovat je do souborů použitelných v počítači - **wav** a **midi**. Dále je možné melodii přehrát na telefonu, pokud to podporuje.

Loga

Loga lze načíst i uložit do mnoha formátů obrázků, mimo jiné i do běžně používaného formátu **bmp**.

Zálohování

Další důležitou částí je zálohování. Zálohovat lze jednak do vlastního formátu *Gammu*, nebo do několika formátů používaných v jiných aplikacích, především PIM (personal information manager – správa osobních údajů):

- **vcs** – vCalendar – výměna událostí z kalendáře a úkolů
- **ics** – iCalendar – výměna událostí z kalendáře a úkolů
- **vcf** – vCard – výměna kontaktů
- **ldif** – výměna kontaktů
- **lmb** – formát používaný aplikací firmy Nokia

Do těchto souborů je možné ukládat i z nich načítat údaje a ty pak nahrát do telefonu.

Vlastní formát *Gammu* je navržen přímo pro účely zálohování, umožňuje tedy uložit veškerá data z telefonu do jediného souboru. Struktura souboru odpovídá běžnému konfiguračnímu souboru **.ini**, a tudíž s ním lze snadno pracovat i z případných dalších aplikací. Vícepoložková data jsou uložena ve dvou položkách – jedna určuje její typ (pro kontakty to například může být číslo mobilu) a druhá pak její hodnotu. Protože by případné unicode znaky mohly působit problémy, existují dvě možnosti jak tyto soubory ukládat – buďto jsou textové položky obsaženy dvakrát, jednou převedené do aktuální znakové sady podle **locales** a podruhé ve formě šestnáctkového výpisu unicode, nebo je kompletně celý soubor uložen v unicode.

3.2.13 Ošetřování chyb

Všechny funkce komunikující s telefonem vrací hodnotu **GSM.Error**, která určuje jestli operace proběhla v pořádku, nebo skončila s nějakou chybou. Chyby jsou členěny velmi

podrobně (v současné době kolem 45 různých chyb), takže je z chyby většinou na první pohled zřejmé, kde nastal problém.

3.3 Moje práce na projektu

Kromě dále uvedeného *python-gammu* a *Wammu*, které *Gammu* používají a které jsem kompletně napsal, jsem i do projektu *Gammu* v rámci diplomové práce přispěl několika částmi:

- kompletní modul pro telefony Alcatel
- vylepšení modulu pro AT telefony – především se jednalo o rozšíření kompatibility s různými telefony a přidání některých nových funkcí
- vylepšení vnitřní struktury projektu – sjednocení sémantiky funkcí pro přístup k různým datům
- zavedení flexibilnějšího formátu pro interní reprezentaci dat – umožňující snadno pracovat s různými daty z různých telefonů (viz 3.2.11)
- podpora operačního systému *Linux* – konkrétně především vylepšení v modulu pro sériový port
- přenositelnost na různé architektury
- vytváření balíčků pro různé distribuce *Linuxu*

Kromě těchto podstatnějších změn jsem provedl i mnoho menších, které jsou podrobněji popsány v changelogu projektu, který je vytištěn jako příloha nebo je vystaven na internetových stránkách *Gammu* - <http://www.mwiacek.com/zips/gsm/gammu/changelog>.

3.3.1 Modul pro telefony Alcatel

Tento modul vychází z mého předchozího projektu pro komunikaci s telefony – *KAlcatel* (viz. <http://www.cihar.com/alcatel>), respektive knihovna *alcasync* (dostupná na stejné adrese). Z důvodu možnosti oddělení vrstvy protokolu od vlastních zpráv zajišťujících komunikaci, bylo nutné zajistit důkladné oddělení těchto částí, čímž se zdrojový kód také značně zpřehlednil.

Protokol

Do protokolu umožňujícím práci s daty v telefonu je možné se přepnout pomocí AT příkazu `AT+CPR0T`. Poté již probíhá komunikace pomocí paketů, které jsou vždy po přijetí potvrzovány.

Tento protokol není dokumentován a výrobce na žádost o dokumentaci reagoval negativně, takže bylo nutné informace o protokolu získat ze stávající aplikace pro *Microsoft Windows*. Tato aplikace naštěstí veškerou komunikaci ukládá do logovacího souboru, takže získat alespoň některé informace bylo celkem snadné. Z těchto údajů již bylo možné vytvořit první implementaci protokolu, která umožňovala komunikovat s telefonem. Během dalšího vývoje byly přidány i další funkce, takže nyní již je možné používat vše co z tohoto protokolu používá aplikace *Alcatelu*.

Pakety jsou číslovány modulo 61 ($0x3d$) a jsou zabezpečeny kontrolním součtem (XOR všech bytů paketu). Každý paket začíná bytem 126 ($0x7e$). Po odeslání paketu jak telefon tak počítač čeká na potvrzení a teprve potom komunikace pokračuje. Potvrzení je zvláštní typ paketu, stejně tak jsou speciální typy pro signalizaci začátku a konce přenosu.

Funkce protokolu

Funkce protokolu se starají o obalení posílaných dat do protokolu, číslování paketů, generování kontrolního součtu, skládání přijatých dat do paketů a ověřování jejich kontrolního součtu. Vyšší vrstva tedy jen posílá a přijímá data a nestará se o nižší část komunikace.

Funkce telefonu

Vlastní komunikace již pak spočívá jen v sestavení dat a následném dekódování. Protože tyto telefony umožňují jak komunikaci pomocí AT příkazů i pomocí vlastního protokolu, je nutné automaticky řešit přepínání těchto režimů a případně použít funkce AT modulu, pokud je to možné.

Telefon v binárním režimu zpřístupňuje kontakty, kalendář a úkoly. Práce se všemi těmito daty je identická, liší se jen v mapování údajů do interních polí. Komunikace se může nacházet v několika stavech:

- attached (připojeno) – je navázáno spojení, ale ještě nebyl vybrán typ dat, se kterými chceme pracovat
- session (sezení) – již je vybrán typ dat (kontakty/kalendář/úkoly), se kterými chceme pracovat
- edit (upravování) – je upravován nějaký záznam. po editaci je potřeba zaznamenat (commit) data do telefonu, jinak nejsou uložena

Čtení probíhá v několika fázích – nejdříve se načte seznam požadovaných záznamů, poté jaké položky konkrétní záznam obsahuje a pak jsou přečteny jednotlivé položky. Veškeré seznamy jsou po přečtení uchovávány v paměti, čímž dochází k značnému urychlení práce.

Úkoly a kontakty se ještě dělí do kategorií. Jejich čtení může probíhat během práce s jednotlivými záznamy, a proto jsou po přečtení uloženy v paměti, aby nebylo nutné pokaždé komunikovat s telefonem.

3.3.2 Modul pro AT příkazy

Kromě mnoha drobných oprav nutných pro funkci různých telefonů, jsem napsal i několik větších změn v tomto modulu:

- zpracovávání rozšířených chybových hlášek od telefonu (CMS a CME chyby)
- podpora pro přidávání nových kontaktů
- práce s různými paměťmi pro textové zprávy

Při vývoji jsem jednak čerpal z norem *ETSI*, týkajících se *AT* příkazů [4, 5, 6, 7, 8, 9, 10, 12, 11], dokumentací některých výrobců [2, 1, 13, 14] a v neposlední řadě také z vlastních zkušeností získaných při vývoji programu pro telefony *Alcatel – KAlcatel*.

Funkce protokolu

Vrstva protokolu je v tomto případě velmi jednoduchá, jediné co zajišťuje je rozsekání dat přicházejících z telefonu na zprávy – každá zpráva obsahuje odpověď na jeden *AT* příkaz, nebo jednu nevyžádanou informaci (například příchozí hovor) od telefonu.

Funkce telefonu

Na této úrovni dochází ke zpracování textových odpovědí a jejich případné uložení do odpovídajících struktur a pokud nastala chyba, její ošetření. Chyby při komunikaci pomocí *AT* příkazů mohou nastat různé: jednak nespecifikovaná chyba – **ERROR** a dále pak **CME ERROR** a **CMS ERROR**. Obě tyto chyby popisují i co je její příčinou a to buďto v numerické nebo textové podobě, v závislosti na nastavení telefonu a jeho vlastnostech. Jak textové tak i numerické chyby jsou správně rozpoznány a nahlášeny uživateli. **CMS** chyby se týkají práce z textovými zprávami, **CME** obecně telefonu nebo komunikace se sítí.

Kapitola 4

Grafické uživatelské rozhraní

Aby byl program použitelný i pro běžné uživatele, musí jít ovládat grafickým uživatelským rozhraním. Z funkcí, které *Gammu* podporuje, jsem se rozhodl vytvořit rozhraní pro kontakty, úkoly, kalendář a textové zprávy, ale s možností případné snadné implementace dalších částí později. Toto rozhodnutí je ovlivněno především možnostmi telefonu, který mám k dispozici, protože ten podporuje právě tyto funkce. Na základě předchozích zkušeností a kladné odezvy uživatelů jsem se rozhodl pro podobné rozvržení rozhraní, jako jsem zvolil u programu *KAlcatel*. Nejdůležitější volbou bylo použití grafické knihovny a programovacího jazyku, protože program by měl být schopný běžet na co největším množství operačních systémů, minimálně pak na *Linuxu* a *Microsoft Windows*.

4.1 Grafická knihovna

S ohledem na použitelnost na *Microsoft Windows* a *Linuxu* (resp. *XWindows*), připadalo v úvahu několik knihoven. Každá má své výhody a nevýhody, které se pokusím shrnout v následujících odstavcích. Vybíral jsem pouze z knihoven, které jsou v rozumné míře používány a lze tedy předpokládat, že s nimi nebude příliš mnoho neočekávaných problémů.

4.1.1 GTK

GTK (<http://www.gtk.org>) knihovna dlouhou dobu existuje pro prostředí *XWindows*, kde původně vznikla pro grafický editor *Gimp*, ale pro *Microsoft Windows* je pořád ještě v ranném stádiu vývoje, i když některé aplikace ji již používají. Právě z důvodu nedokončeného vývoje verze pro *Microsoft Windows*, jsem se rozhodl tuto knihovnu nepoužít.

Kromě grafické verze *GTK*, existuje ještě verze fungující v konzoli, ale to je pro tento program nepodstatné.

4.1.2 Qt

Knihovna *Qt* (<http://www.trolltech.com>) funguje na obou požadovaných platformách, její nevýhodou je potřeba komerční licence pro kompilaci na *Microsoft Windows*. V současné době se jedná o jednu z nejpoužívanějších knihoven, především díky velkému rozvoji desktopu *KDE* (<http://www.kde.org>), který je na této knihovně postaven.

4.1.3 wxWindows

Knihovna *wxWindows* (<http://www.wxwindows.org>) je volně šiřitelná a podporuje obě požadované platformy. Oproti oběma předchozím zobrazuje na každé platformě rozhraní použitím nativních prvků na dané platformě. Především z licenčních důvodů jsem se rozhodl pro tuto knihovnu místo *Qt*.

Nevýhodou *wxWindows* v současné době je používání staršího *GTK* 1.2 (toto se týká jen verze pro *Linux*), které nenabízí tak pěkný vzhled jako *GTK* 2. Toto je však problém pouze dočasný, protože *wxWindows* *GTK* 2 již podporují a je jen otázka času, kdy distribuce *Linuxu* toto aktivují. Samozřejmě se tato změna nijak nedotkne zdrojových kódů aplikací.

wxWindows abstrahuje od vlastního toolkitu použitého pro zobrazování a tím dosahuje nativní vzhled na všech podporovaných platformách. V současné době jsou podporovány tyto platformy - *Microsoft Windows*, *MacOS* a platformy na kterých lze používat *GTK+* a *Motif*.

4.2 Programovací jazyk

Vybraná knihovna *wxWindows* umožňuje programování v *C++*, *Pythonu* nebo *Perlu*. Protože moje programátorské zkušenosti s *Perlem* jsou minimální, tak jsem tento jazyk z uvažovaných možností vyřadil velmi záhy. Naopak jak s *C++* i s *Pythonem* mám zkušenosti z tvorby několika projektů.

Výhodou *C++* je o trochu rychlejší běh aplikací a odpadá nutnost instalovat interpret jazyka *Python*. Vývoj v *Pythonu* je naopak rychlejší, protože jednak odpadá mnohdy zdlouhavá kompilace a díky garbage collectoru se člověk nemusí starat o uvolňování objektů, ty jsou automaticky uvolněny, jakmile nejsou nikde používány.

Já jsem zvolil *Python*, protože výhody které to přináší, převýší nevýhody. Instalace interpretu na systém *Linux* není obvykle nutná, protože naprostá většina současných distribucí *Python* obsahuje v základní instalaci, na *Microsoft Windows* je potřeba *Python* instalovat, ale to lze obejít sloučením všech potřebných modulů, kódy a interpreteru do jednoho spustitelného souboru, takže uživatel instaluje jen jeden program.

Aby bylo možné komunikovat s *Gammu* z *Pythonu*, bylo nutné vytvořit mezi nimi rozhraní. Toto rozhraní jsem nazval *python-gammu* a je popsáno v následující kapitole.

Kapitola 5

python-gammu

Projekt *python-gammu* umožňuje používat *Gammu* z *Pythonu*. Jeho snahou není přidávat jakoukoliv novou funkcionalitu, ale jen převádět datové struktury *Gammu* na objekty v *Pythonu* a naopak a zpřístupnit funkce *Gammu*.

Aktuální verze *python-gammu* včetně zdrojových kódů a balíčků pro vybrané operační systémy bude vždy dostupná na <http://www.cihar.com/gammu/python>. Program byl uveřejněn pod licencí *GNU GPL*.

5.1 Reprezentace dat

Pro struktury *Gammu* bylo nutné zvolit nějaký objekt v *Pythonu*. V úvahu přicházely v podstatě dvě možnosti – pro každou strukturu vytvořit specifický objekt nebo veškeré hodnoty ukládat pomocí standardních objektů dostupných v *Pythonu* – seznam (list) a slovník (dictionary). Já jsem zvolil druhou možnost, především kvůli snadnosti implementace a práce s takto uloženými hodnotami.

Veškeré hodnoty (jak vstupní tak i výstupní) tedy jsou ukládány do slovníků, případně jejich seznamů. Pro klíče slovníku jsou použity stejné názvy, které používá *Gammu* ve strukturách pro odpovídající data. Z důvodů přebytečnosti jsou však některé atributy odstraněny (například počet položek v seznamu). Některá data (například bitmapa obrázků) jsou přístupné v úplně jiné podobě, protože původní byla příliš orientována na interní zpracování a nikoli jako data, která by měla být případně přístupná dalším modulům. Hodnoty různých výčtových typů jsou převáděny na text, jen jejich případná předpona je odstraněna - například místo `SMS_FORMAT_Text` se použije jen `Text`, protože z kontextu je zřejmé, čeho se hodnota týká.

5.2 Struktura modulu

Modul obsahuje jedinou třídu - `StateMachine`, která obaluje strukturu `GSM.StateMachine` z *Gammu*, jen nenabízí přímý přístup k jejím hodnotám. Tato třída má metody, které odpovídají funkcím modulu telefonu a některé inicializační a ukončovací funkce, které jsou v *Gammu* samostatné.

Dále modul obsahuje některé funkce. Jedná se především o podporu pro kódování a dekodování zpráv, ukládání a načítání záloh, získání informací o používané knihovně a nastavení globálních parametrů *Gammu*.

5.3 Výjimky

Jednou z největších změn oproti *Gammu* v C je použití výjimek, čímž se zjednoduší obsluha chybových stavů. Základní třídou pro všechny výjimky související s komunikací s telefonem je obecná chyba `gammu.GSMError`. Potomci této třídy již pak přímo reprezentují chyby *Gammu* a to i se stejnými názvy, jako jsou použity v *Gammu*.

5.4 Funkce Gammu

V *python-gammu* zatím nejsou přístupné všechny funkce *Gammu*. Jejich rozsah je zatím omezen na funkce, které mám možnost otestovat na svém telefonu, případné přidání dalších funkcí je velmi snadné, většinou je potřeba implementovat jen konverzi příslušných dat.

Funkce, které vyžadují připojení k telefonu nebo ho ovlivňují, jsou obsaženy v třídě `StateMachine`. Tyto funkce lze rozdělit do několika hlavních oblastí:

- změna nastavení
- inicializace a ukončení připojení
- zjištění informací o telefonu, síti a podobně
- přečtení a nastavení data a budíku
- manipulace s kategoriemi úkolů a kontaktů
- manipulace s kontakty
- manipulace se složkami textových zpráv
- manipulace s textovými zprávami
- manipulace s úkoly

- manipulace s událostmi v kalendáři
- nastavování ladicích informací připojení

Funkce, které připojení nevyžadují ani ho přímo neovlivňují, jsou přístupné přímo z modulu `gammu`, jsou to následující druhy funkcí:

- spojování textových zpráv
- dekódování textových zpráv do programem snadno zpracovatelného formátu
- kódování textových zpráv do *PDU* dat
- nastavování globálních ladicích informací
- ukládání melodií
- ukládání a načítání záloh

Kapitola 6

Wammu

Wammu je grafické uživatelské rozhraní postavené na *python-gammu*. Je napsané za použití grafické knihovny *wxPython*, která zajišťuje přenositelnost na mnoho platforem.

Aktuální verze *Wammu* včetně zdrojových kódů a balíčků pro vybrané operační systémy bude vždy dostupná na <http://www.cihar.com/gammu/wammu>. Program byl uveřejněn pod licencí *GNU GPL*.

6.1 Vzhled a jeho implementace

Vzhled aplikace je z velké části inspirován projektem *KAlcatel*, který je vlastně předchůdcem *Wammu*. Jak *KAlcatel* vypadá, je vidět na obrázku 6.1. Tento vzhled byl velmi kladně přijat uživateli a proto jsem se rozhodl použít velmi podobný i pro *Wammu*.

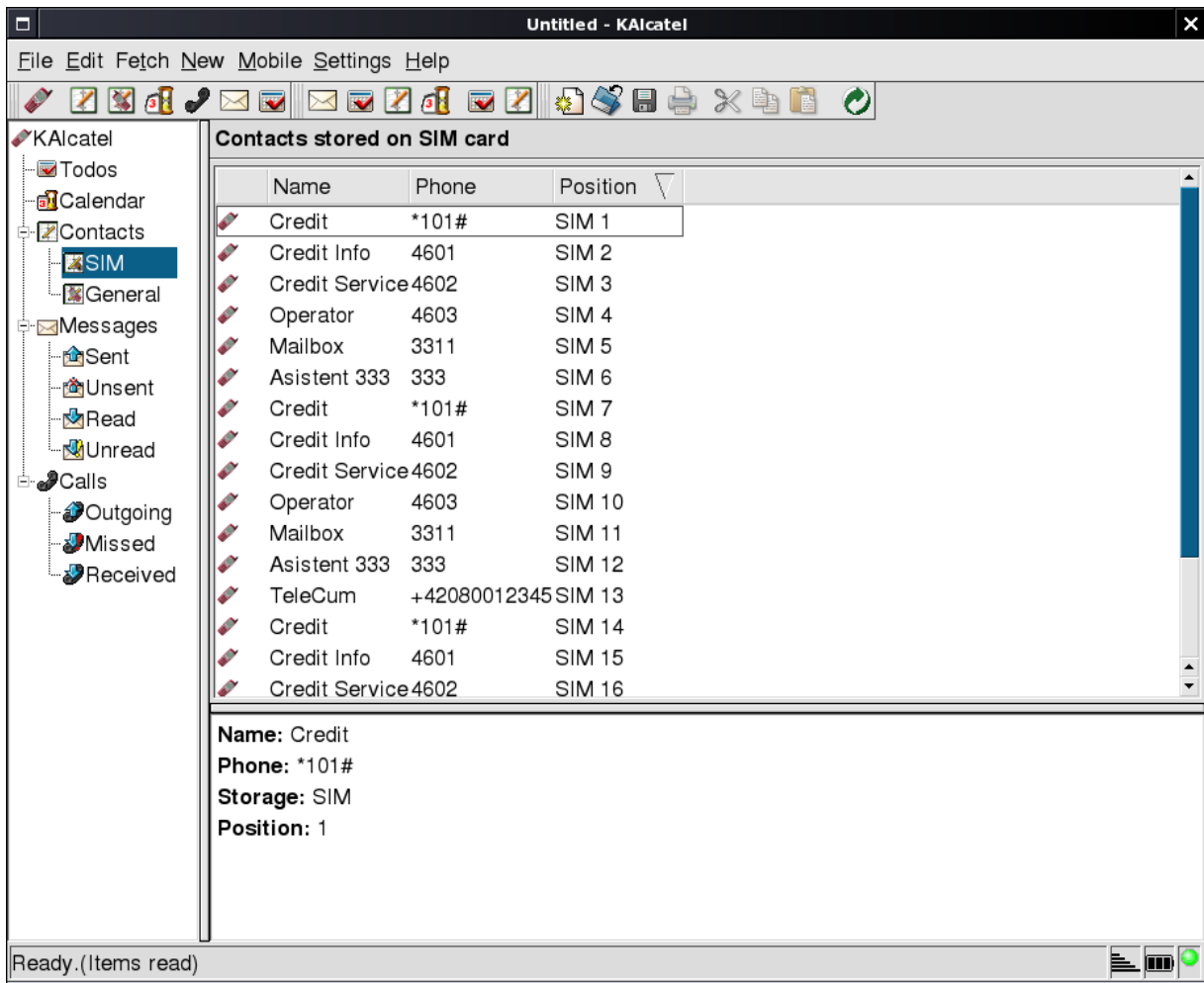
Základní struktura je velmi jednoduchá - vlevo je zobrazena stromová struktura dostupných typů objektů, vpravo nahoře jejich výpis a vpravo dole detailní zobrazení aktuálně vybraného objektu, výsledek je vidět na obrázku 6.2.

Většina částí základního rozhraní má celkem jednoznačně určeno jak bude implementována (například strom objektů), u jiných je naopak mnoho možností jak dosáhnout podobné funkcionality (například zobrazení a editace objektů). Této druhé části, tedy „problematičtější“, rozhraní se zde budu věnovat podrobněji.

6.1.1 Zobrazení objektů

Pro zobrazení objektů je jednak možné použít kombinaci různých statických prvků, které *wxPython* nabízí (text, obrázek, ...), nebo objekty převést do nějakého strukturovaného jazyka, který potom zobrazit ve specializovaném prvku. *wxPython* přímo pro tyto účely nabízí *wxHTML*, což je jakýsi zjednodušený prohlížeč *HTML*, do kterého je ale možné pomocí speciálních tagů vkládat i další prvky, například tlačítka a podobné.

Možnost použít *wxHTML* mi připadala optimální, protože uživatel pak bude mít mož-

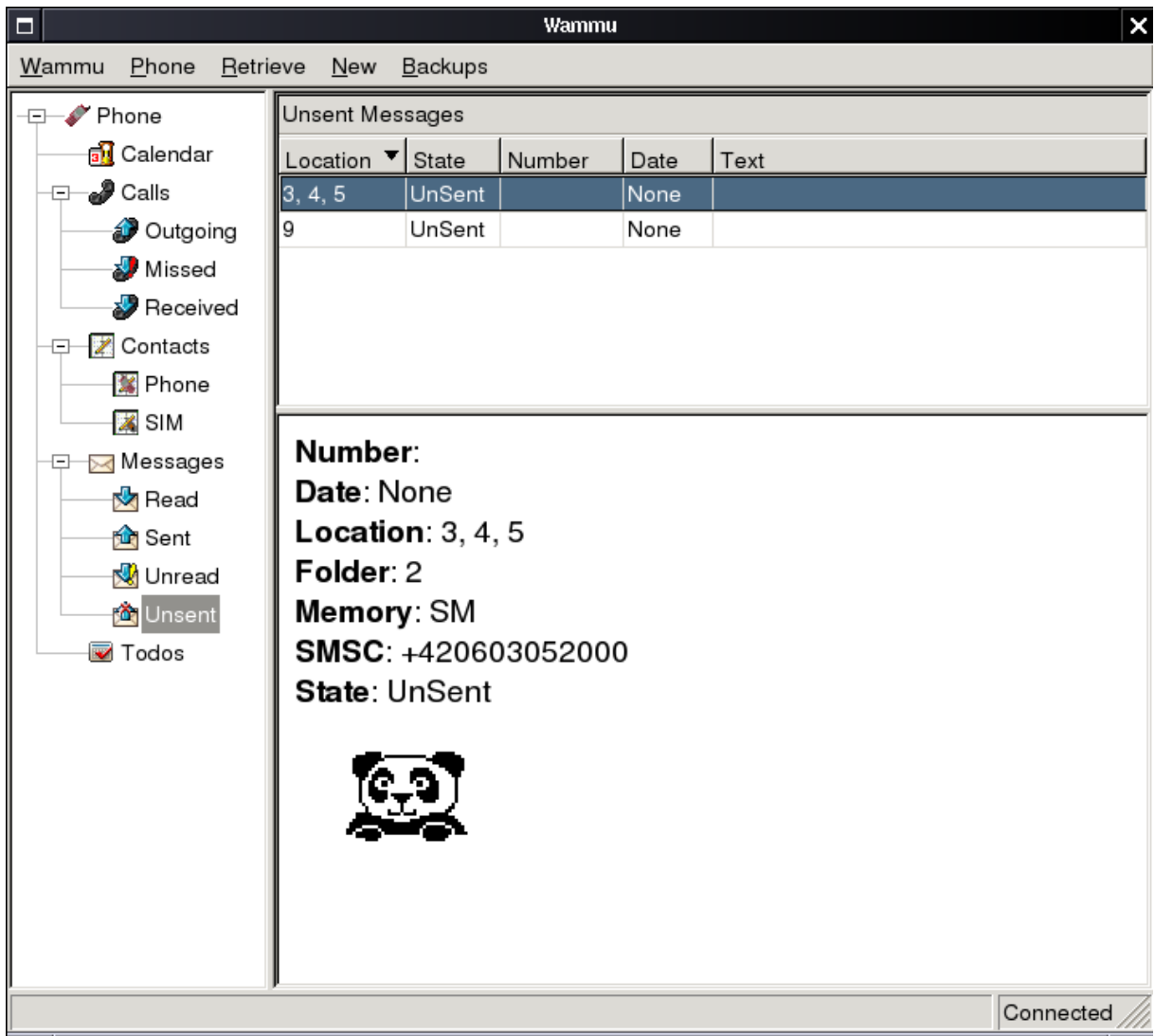


Obrázek 6.1: Vzhled programu KAlcatel

nost snadno text zkopírovat a použít ho i v jiné aplikaci. Bohužel text z `wxHTML` v současné stabilní verzi `wxWindows` (2.4.2.4) kopírovat nelze, to bude umožňovat až novější řada (2.5). I přes tuto drobnou nevýhodu jsem `wxHTML` použil, protože v případě použití samostatných prvků by kopírování bylo ještě komplikovanější.

6.1.2 Editace kontaktů, úkolů a událostí kalendáře

Protože všechny tyto objekty jsou reprezentovány podobnými strukturami, rozhodl jsem se i jejich editaci postavit na jediné třídě. Každý z těchto objektů může mít proměnlivý počet hodnot různého typu, což musí editační třída ošetřit. Toto jsem vyřešil seznamem editačních komponent, která každá obstarává jednu hodnotu. Jak editor ve výsledku vypadá je vidět na obrázku 6.3.



Obrázek 6.2: Vzhled programu Wammu

6.1.3 Editace zpráv

Editor zpráv musí umožňovat skládání zpráv z různých částí tak, aby výsledkem mohly být například *EMS* zprávy. Kromě základních ovládacích prvků obsahuje okno editoru tři části - nalevo seznam aktuálních částí zprávy, napravo seznam všech dostupných částí (respektive těch, pro které je implementována editace) a dole vlastní editace vybrané aktuální části. Toto uspořádání umožňuje snadno skládat zprávy z mnoha částí jako obrázky, melodie a podobně. Na obrázku 6.4 je vidět, jak může editace zprávy vypadat.

V současné době je možné upravovat textové zprávy včetně formátování a předdefinované animace a melodie. Položky, které nemohou být upravovány, však lze bez problémů nezměněné přeposlat.

The screenshot shows a dialog box titled "Editing contact ME:27" with a close button (X) in the top right corner. The dialog contains the following fields:

- Location (0 = auto): 27
- Memory type: ME
- 1. Text_FirstName: SOFTCOM computers
- 2. Text_Company: SOFTCOM computers
- 3. Category: 4
- 4. Private:
- 5. Number_Work: +420222514400
- 6. Number_General: +420222514402
- 7. Number_Fax: +420222514403
- 8. Number_Mobile: +420604163999
- 9. Number_Home: +420222514401
- 10. Text_Email: sales@computershop.cz

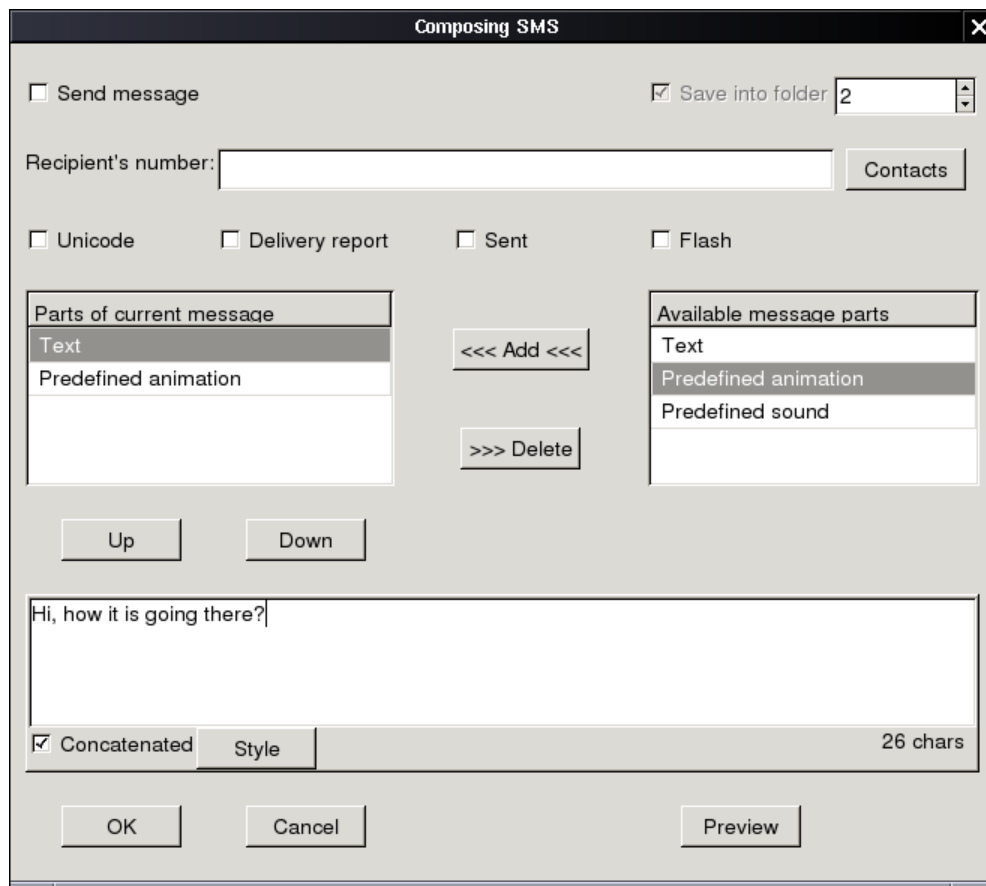
At the bottom of the dialog are three buttons: OK, Cancel, and More.

Obrázek 6.3: Editace kontaktu

6.1.4 Zobrazení zpráv

Velký důraz jsem kladl na práci se zprávami, protože to je jedna z nejvíce používaných služeb na mobilních telefonech. Kromě již popsaného modulárního editoru je důležité i jejich zobrazení. Díky použité komponentě wxHTML se velmi snadno mění formátování textu. Kromě toho je možné zobrazit i obrázky a animace (jak je vidět na obrázku 6.2) a po kliknutí umožňuje přehrát melodie.

Pro přehrávání melodií jsem poměrně dlouho vybíral vhodný způsob. V úvahu přicházela přímá konverze na audio data a ty přehrát, což je při požadavku kvalitního zvuku poměrně náročný úkol, nebo vytvoření nějakého formátu, který bude schopný přehrát jiný program. Nakonec zvítězila konverze do formátu *midi* (již dříve implementovaná v *Gammu*) a poté jeho přehrání pomocí externího přehrávače, protože tato metoda poskytuje nejlepší výsledný zvuk. Pro vlastní přehrávání jsem zvolil program *timidity*, který pomocí vlastní sady vzorků vytváří zvuk a přehrávání tudíž není závislé na použité zvukové kartě. Další informace o programu je možné získat například z jeho stránek – <http://www.goice.co.jp/member/mo/timidity/>.



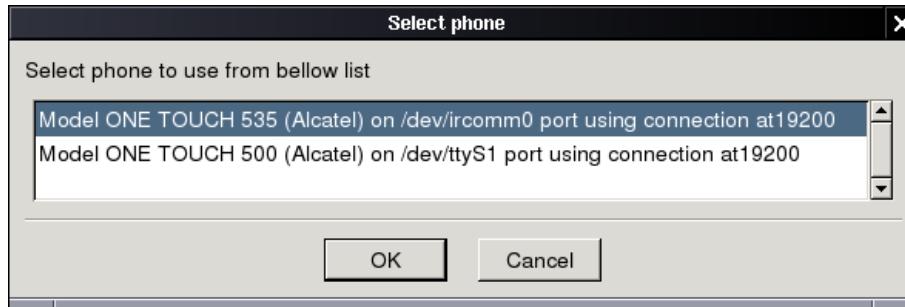
Obrázek 6.4: Editace zprávy

6.2 Komunikace s telefonem

Z důvodů možné dlouhé doby trvání, je komunikace s telefonem prováděna ve vlastním vlákně. Komunikace mezi vlánky je zajišťována pomocí událostí wxPythonu, které je možné posílat mezi jednotlivými vlákny a občasná synchronizace vláken je řešena použitím zámků. Tímto návrhem je zajištěno, že rozhraní bude reagovat, i když komunikace s telefonem z jakéhokoliv důvodu bude trvat velmi dlouho.

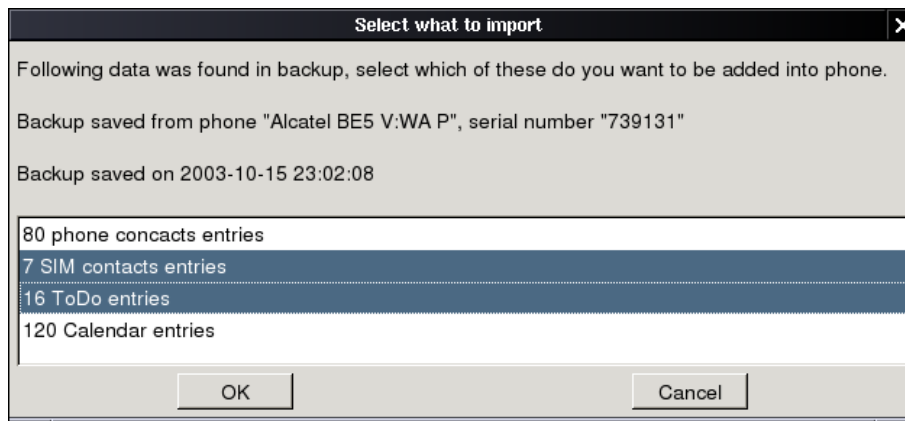
6.3 Vyhledávání telefonu

Protože pro mnoho uživatelů může být problém úvodní nastavení aplikace. Pro tyto účely byla vytvořena funkce pro vyhledávání telefonu. Ta vytvoří mnoho vláken a v každém z nich zkusí různá připojení na jedno zařízení. Tímto způsobem lze poměrně rychle projít běžně používané sériové a IrDA porty a uživatel pak již jen zvolí některý z nalezených telefonů a nemusí se zajímat o pojmenování portu a způsob připojení. Výsledek vyhledávání je vidět na obrázku 6.5.



Obrázek 6.5: Výsledek vyhledávání telefonu

6.4 Export a import dat



Obrázek 6.6: Nabídka importu

Jedním z úkolů bylo také umožnit snadnou výměnu dat mezi různými telefony. To je dosaženo pomocí exportu a importu dat, které kromě toho umožňuje i jednoduchou synchronizaci s dalšími programy. Podporovány jsou všechny formáty, které podporuje *Gammu* (viz 3.2.12). Exportovat je možné jak všechna data načtená z telefonu, tak i jednotlivé položky. Při importu si uživatel zvolí, který typ položek chce importovat (viz obrázek 6.6) a tyto jsou pak přidány do telefonu.

6.5 Lokalizace

Program je možné lokalizovat pomocí standardních nástrojů používaných pro lokalizaci (*gettext* apod.). Já jsem provedl překlad rozhraní do češtiny, takže program s uživatelem komunikuje česky, pokud má nastavené české locales.

6.6 Požadavky

Wammu pro svůj běh potřebuje *python-gammu*, které zajišťuje komunikaci s telefonem prostřednictvím *Gammu*, *wxPython* (minimálně verze 2.4.1.2) použitý pro rozhraní a v případě, že chce uživatel přehrávat melodie, tak ještě program *timidity*.

Skutečnost, že je program napsán v *Pythonu* neznamená pro dnes používané počítače žádné znatelné zpomalení. To je způsobeno především tím, že naprostá většina funkcí je implementována v C (vlastní komunikace s telefonem i komponenty uživatelského rozhraní) a tyto části jsou vlastně v *Pythonu* „jen“ pospojovány. Kromě toho jsou všechny části napsané v *Pythonu* na uživatelově systému předkompilované do byte kódu, čímž se zrychlí především start aplikace.

Kapitola 7

Synchronizace

Mým úkolem bylo také zjistit možnosti synchronizace telefonu s *PIM* software v počítači. *PIM* je zkratka pro *Personal Information Manager*, jedná se obvykle aplikace obsahující adresář a kalendář, i když mnohdy to jsou samostatné programy, nebo naopak jsou integrovány ještě s dalšími funkcemi, velmi často s emailovým a news klientem.

O tuto oblast jsem se zajímal pouze na systému *Linux*, protože pro synchronizaci je potřeba mít přístupné rozhraní programu a to je pro naprostou většinu programů pro *Microsoft Windows* přístupné pouze v rámci placeného vývojového kitu (*SDK*).

7.1 Problémy

Při synchronizaci s většinou telefonů nastává problém nemožnosti přímého zjištění jaké záznamy byly v telefonu změněny. Toto se dá buďto porovnáváním kompletních záznamů se záznamy v *PIM* aplikaci, což zbytečně zatěžuje uživatele, protože musí manuálně řešit konflikty, které ve skutečnosti vůbec nenastaly, nebo zálohováním předchozího stavu údajů v telefonu do souboru na disku a vygenerování rozdílů mezi těmito uloženými daty a aktuálními daty v telefonu. Toto řešení má také své nevýhody – především v případě synchronizace dat na vzdáleném počítači (pokud jsou například uloženy v *LDAP* databázi) je potřeba vždy synchronizaci provádět se stejnými lokálními daty.

7.2 Existující projekty

Projektů snažících se o synchronizaci existuje v současné době několik. Největší část podobných aplikací je psána pro počítače *Palm* a *Psion*, pro telefony existují v podstatě jenom dva projekty v různém stádiu vývoje.

7.2.1 KnokiiSync

Jedná se o projekt pro synchronizaci mezi *Gnokii* a *KAddressbookem* napsaný pro prostředí *KDE*. Bohužel projekt není napsán modulárně, takže není snadné ho rozšířit o další aplikace. Stránky o tomto projektu naleznete na <http://knokiisync.sourceforge.net>.

7.2.2 Multisync

Multisync (<http://multisync.sourceforge.net>) je určen pro obecnou synchronizaci, je navržen modulárně, takže je možné snadno napsat pluginy pro další aplikace. V současné době obsahuje následující moduly:

Zálohování Zálohování do textového souboru

Ximian Evolution Synchronizace s *PIM Ximian Evolution*

IrMc Synchronizace s telefony podporujícími *OBEX*, komunikace probíhá přes kabel, *IrDA* nebo *Bluetooth*

Opie Synchronizace s počítači používajícími systém *Opie*

Palm Synchronizace s počítači *Palm*

SyncML Synchronizace se servery a klienty *SyncML*

7.3 Shrnutí

Z pohledu na existující projekty se jako nejlepší jeví rozšířit schopnosti *Multisyncu* o další telefony. Bohužel tento projekt má nepříliš dokumentované rozhraní, a proto je psaní modulu poměrně náročné. Především z tohoto důvodu jsem od implementace synchronizace upustil.

Přesto však je spolupráce s různými programy možná za použití exportu a importu ve *Wammu* (viz 6.4), ale tato funkce je vhodná spíše pro jednorázový přenos dat jedním či druhým směrem, než pro plnohodnotnou synchronizaci.

Kapitola 8

Zhodnocení

Cílem této práce bylo vytvořit grafickou aplikaci pro práci s daty v mobilním telefonu.

Výsledek je aplikace postavená na již dříve existujícím projektu *Gammu*, na jehož vývoji jsem se značnou měrou podílel. Vytvořená aplikace *Wammu* je napsaná pomocí platformově nezávislé knihovny *wxWindows*, respektive jejího rozhraní v programovacím jazyce *Python* – *wxPython*. Toto by mělo umožňovat přenositelnost i na jiné platformy než je Linux.

V praxi jsem ověřil funkčnost jen na několika distribucích *Linuxu* na různých architekturách (x86, AMD64, PPC a IA64). Na *Microsoft Windows* jsem program nezkoušel, protože mám s tímto systémem minimální zkušenosti a jeho příprava pro možnost kompilace by mi zabrala neúměrné množství času. Nicméně nepředpokládám, že by nastaly nějaké problémy, protože samotné *Gammu* na této platformě funguje, stejně tak není žádný problém s *Pythonem* ani grafickou knihovnou *wxPython*, takže případné drobné komplikace by mohly nastat při kompilaci modulu *python-gammu*.

První verze *Wammu* byla zveřejněna začátkem ledna, během několika dnů jsem zaznamenal stovky stažení a mnoho pozitivních ohlasů i ohlášení nalezených chyb. Chyby byly opraveny v následujících verzích, ale přesto program určitě není bezchybný. To je způsobeno mimo jiné i širokou podporou různých zařízení a tudíž nemožností vyzkoušet chování se všemi kombinacemi zařízení, operačních systémů a dat.

Z ohlasu uživatelů je zřejmé, že se jednalo o program, který mnoha uživatelům telefonu a počítače chyběl a určitě najde uplatnění i v budoucnosti, pokud se podaří udržovat současný trend podpory nových telefonů. Kromě kladného ohlasu uživatelů mě potěšila i skutečnost, že program byl velmi rychle zařazen i do některých *Linuxových* distribucí. Projekt pro mě rozhodně nekončí datem odevzdání diplomové práce, ale hodlám ho vyvíjet i nadále.

Literatura

- [1] ALCATEL. *Alcatel One Touch's 535 Data Reference Manual*, verze 1.0, 2002.
- [2] ALCATEL. *Alcatel One Touch's 715 Data Reference Manual*, verze 1.0, 2002.
- [3] ASKEY. *Příručka pro uživatele FAX/DATA/VOICE modem*, verze 5/2001, 2001.
- [4] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 02.30*, verze 5.0.0, 1995. Digital cellular telecommunications system (Phase 2+); Man-Machine Interface (MMI) of the Mobile Station (MS).
- [5] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 02.82*, verze 5.0.0, 1995. Digital cellular telecommunications system (Phase 2+); Call Forwarding (CF) supplementary services - Stage 1.
- [6] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 03.38*, verze 5.3.0, 1996. Digital cellular telecommunications system (Phase 2+); Alphabets and language-specific information.
- [7] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 03.40*, verze 5.1.0, 1996. Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point-to-Point (PP).
- [8] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 03.41*, verze 5.3.0, 1996. Digital cellular telecommunications system (Phase 2+); Technical realization of Short Message Service Cell Broadcast (SMSCB).
- [9] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 04.08*, verze 5.3.0, 1996. Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification.
- [10] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 04.11*, verze 5.1.0, 1996. Digital cellular telecommunication system (Phase 2+); Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface.

-
- [11] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 07.07*, verze 5.0.0, 1996. Digital cellular telecommunications system (Phase 2+); AT command set for GSM Mobile Equipment (ME).
- [12] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE. *GSM Technical Specification 07.05*, verze 5.5.0, 1998. Digital cellular telecommunications system (Phase 2+); Use of Data Terminal Equipment - Data Circuit terminating; Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS).
- [13] SIEMENS. *AT Command Set for SIEMENS Mobile Phones S35i, C35i, M35i*, verze 1.0, 2000.
- [14] SIEMENS. *AT command set for S45 Siemens mobile phones and modems*, verze 1.8, 2001.

Rejstřík

A

Alcatel manažer, 5
All Phone Data Suite, 5
APDS, *viz* All Phone Data Suite
AT příkazy, 1

B

Bluetooth, 1

E

EMS, *viz* Enhanced Messaging Service
Enhanced Messaging Service, 2
ETSI, *viz* European Telecommunications
Standards Institute
European Telecommunications Standards
Institute, 2

G

Gammu, *viz* GNU All Phone Management
Utilities
funkce, 9
modul protokolu, 7
modul telefonu, 8
Alcatel, 8
AT, 8
Nokia, 8
OBEX, 8
Symbian, 9
modul zařízení, 7
struktura, 6
Global System for Mobile Communicati-
ons, 1
Gnokii, 3
GNU All Phone Management Utilities (Ga-
mmu), 3, 6

GSM, *viz* Global System for Mobile Com-
munications

GSM Technical Specification, 1

Gsmlib, 4

GTS, *viz* GSM Technical Specification

H

Handset Manager, 5

I

Infrared Data Association, 2

International Telecommunications Union
, 2

IrDA, *viz* Infrared Data Association

K

KAlcatel, 4, 26

Kandy, 4

KSiemens, 4

O

OBEX, *viz* Object Exchange

Object Exchange, 2

Oxygen Phone Manager, 5

P

PDU, *viz* Protocol Data Unit

Protocol Data Unit, 2

python-gammu, 22

S

Short Message Service, 2

Smart Messaging, 2

SMS, *viz* Short Message Service

W

Wammu, 25

Wireless Application Protocol, 2