

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## OVLÁDÁNÍ DESKTOPOVÉ PREZENTACE POMOCÍ MOBILNÍHO TELEFONU S BLUETOOTH

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MIROSLAV TESAŘ

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# OVLÁDÁNÍ DESKTOPOVÉ PREZENTACE POMOCÍ MOBILNÍHO TELEFONU S BLUETOOTH

WIRELESS CONTROL OF DESKTOP PRESENTATION USING

MOBILE PHONE WITH BLUETOOTH

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MIROSLAV TESAŘ

VEDOUCÍ PRÁCE

SUPERVISOR

ING. PETR NOVOSAD

BRNO 2009

## **Abstrakt**

Cílem projektu je osvobodit osobu provádějící prezentaci z notebooku nebo i stolního počítače od nutnosti ovládat prezentaci z klávesnice. Systém využívá mobilní telefon s podporou mobilní Javy a Bluetooth technologie ke komunikaci s počítačem. Práce se zaměřuje na užívání v širokém spektru mobilních telefonů a je nezávislá na prezentačních programech možností samostatné definice ovládacích prvků uživatelem. Dokument také obsahuje krátké seznámení s technologiemi mobilní Javy a Bluetooth.

## **Abstract**

The project aims to relieve a person performing a notebook or desktop computer based presentation of a need to control the presentation by a keyboard. The system uses a mobile phone with mobile Java and Bluetooth technology to communicate with the computer. The project focuses on the use by a wide range of mobile phones and it is independent of the presentation program using separate definition of user controls. This document also contains a brief introduction to mobile Java technology and Bluetooth.

## **Klíčová slova**

Bluetooth, J2ME, Java, Prezentace, Mobilní telefon.

## **Keywords**

Bluetooth, J2ME, Java, Presentation, Mobile phone.

## **Citace**

Miroslav Tesař: Ovládání desktopové prezentace pomocí mobilního telefonu s Bluetooth, bakalářská práce, Brno, FIT VUT v Brně, 2009

# Ovládání desktopové prezentace pomocí mobilního telefonu s Bluetooth

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petra Novosada.

.....

Miroslav Tesař  
19. května 2009

## Poděkování

Děkuji panu Ing. Novosadovi za odbornou pomoc a připomínky při zpracování práce.

© Miroslav Tesař, 2009.

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

# Obsah

|          |                                 |           |
|----------|---------------------------------|-----------|
| <b>1</b> | <b>Úvod</b>                     | <b>3</b>  |
| <b>2</b> | <b>Bluetooth</b>                | <b>4</b>  |
| 2.1      | Specifikace Bluetooth           | 4         |
| 2.2      | Rozhraní Bluetooth              | 4         |
| 2.3      | Protokoly                       | 5         |
| 2.4      | Profily                         | 6         |
| <b>3</b> | <b>J2ME</b>                     | <b>8</b>  |
| 3.1      | Konfigurace                     | 8         |
| 3.2      | Virtuální stroje                | 9         |
| 3.3      | Profily                         | 10        |
| 3.4      | J2ME a Bluetooth                | 11        |
| <b>4</b> | <b>Neformální specifikace</b>   | <b>12</b> |
| 4.1      | Požadavky na hardware           | 12        |
| 4.2      | Požadavky na aplikaci           | 12        |
| 4.3      | Požadavky na funkčnost          | 12        |
| <b>5</b> | <b>Analýza</b>                  | <b>14</b> |
| 5.1      | Bezdrátová komunikace           | 14        |
| 5.2      | Softwarové vybavení telefonu    | 14        |
| 5.3      | Programovací jazyky a Bluetooth | 15        |
| 5.4      | Prezentační programy            | 15        |
| 5.5      | Existující zařízení             | 16        |
| <b>6</b> | <b>Návrh</b>                    | <b>17</b> |
| 6.1      | Mobilní telefon                 | 17        |
| 6.2      | Počítač                         | 19        |
| <b>7</b> | <b>Implementace</b>             | <b>21</b> |
| 7.1      | Komunikace pomocí Bluetooth     | 21        |
| 7.2      | Serverová část                  | 21        |
| 7.3      | Klientská část                  | 24        |
| <b>8</b> | <b>Závěr</b>                    | <b>27</b> |
| 8.1      | Shrnutí                         | 27        |
| 8.2      | Budoucí vývoj                   | 27        |

|          |                                |           |
|----------|--------------------------------|-----------|
| <b>A</b> | <b>Obsah CD</b>                | <b>30</b> |
| <b>B</b> | <b>Uživatelská příručka</b>    | <b>31</b> |
|          | B.1 Než začneme . . . . .      | 31        |
|          | B.2 Spuštění serveru . . . . . | 31        |
|          | B.3 Spuštění klienta . . . . . | 32        |
|          | B.4 XML soubor . . . . .       | 33        |
| <b>C</b> | <b>Příklad XML souboru</b>     | <b>35</b> |

# Kapitola 1

## Úvod

V dnešní době je velké množství odborných nebo populárně naučných výkladů přednášejícími podpořena prezentacemi na počítači. Téměř každý z nás si již minimálně jednou vytvořil prezentaci a nebo vyzkoušel, jak se prezentace ovládá.

Ve většině případů je vždy nutné být fyzicky přítomen u počítače k provedení přechodu na další snímek. Existují nástroje, které přednášejícího této nutnosti zbavují, ale často se jedná o specializované nástroje.

Cílem této práce je tuto „svobodu“ přinést co nejširšímu počtu uživatelů. Mobilní telefon má dnes v kapse téměř každý. Uživateli tedy stačí vlastnit mobilní telefon s podporou Javy pro malá zařízení (J2ME) a bezdrátové komunikační technologie Bluetooth. Počítač na kterém prezentace bude probíhat musí být vybaven operačním systémem Windows XP, nainstalovanou Javou a bezdrátovým rozhraním Bluetooth.

Ve druhé kapitole je popsána technologie Bluetooth, její základní rysy, parametry, rozdělení a protokoly. Téma Bluetooth je velice rozsáhlé, a proto jsou zde popsány pouze základy.

Třetí kapitola popisuje další důležitý stavební prvek celé práce a to J2ME, její konfigurace, profily a základní třídy, na kterých bude mobilní část aplikace stavět.

Čtvrtá kapitola obsahuje neformální specifikaci, tedy rozebrání požadavků na systém.

Pátá kapitola popisuje analýzu problému a dostupné možnosti, jak by mohl být problém řešen.

V šesté kapitole je popsán návrh systému, tedy jak bude aplikace vypadat, co bude umět a jakým způsobem toho dosáhneme.

V sedmé kapitole je popsáno, jak probíhala samotná implementace systému, a co která z jeho částí dělá.

Poslední osmá kapitola obsahuje závěrečné shrnutí a možný budoucí vývoj práce.

## Kapitola 2

# Bluetooth

Bluetooth[1] je bezdrátová komunikační technologie nahrazující propojení mezi dvěma a více elektrickými zařízeními kabelem na krátkou vzdálenost. Pro svou malou energetickou náročnost a díky malým rozměrům adaptéru je Bluetooth využíváno v různých zařízeních (osobní počítač, mobilní telefon, tiskárny a další).

Bluetooth technologie je definována standardem IEEE 802.15.1 a patří do kategorie PAN (Personal Area Network) a její název je odvozen z anglického jména dánského krále Haralda II. „Modrozuba“ (údajná záliba v borůvkách), který využil svých diplomatických schopností k sjednocení skandinávského lidu. I technologie Bluetooth usnadňuje komunikaci mezi zařízeními a je tedy „novodobým technologickým“ sjednotitelem.

### 2.1 Specifikace Bluetooth

Jak již bylo řečeno dříve, Bluetooth je definováno ve standardu IEEE 802.15.1 a patří do kategorie PAN. Vyskytuje se dnes již v pěti vývojových verzích (1.0, 1.2, 2.0, 2.1 a nejnovější 3.0). Nejnovější verze 3.0 zvyšuje přenosovou rychlost ze současných 3Mb/s (Bluetooth 2.1) na 24Mb/s. Nové přístroje s touto specifikací se objeví až v roce 2010. V dnešní době je stále ještě využíváno Bluetooth specifikace 2.1.

### 2.2 Rozhraní Bluetooth

Bluetooth pracuje v bezlicenčním pásmu 2,4 GHz (stejně jako Wi-Fi). K přenosu využívá metodu rozprostřeného spektra s přeskokováním kmitočtů (FHSS), při němž se během jedné sekundy radiový signál 1600-krát přepne (přeladí) mezi 79 frekvencemi v rozestupu 1MHz. Toto pomáhá zvýšit odolnost spojení vůči rušení na stejné frekvenci.

Každé Bluetooth zařízení spadá do jedné z následujících tříd, jak ukazuje tabulka 2.1.

| Kategorie | Maximální povolený výkon [mW] | Dosah (teoretický) [m] |
|-----------|-------------------------------|------------------------|
| Třída 1   | 100                           | 100                    |
| Třída 2   | 10                            | 10                     |
| Třída 3   | 2,5                           | 1                      |

Tabulka 2.1: Rozdělení Bluetooth do tříd.



Mobilní telefony nejčastěji spadají do druhé třídy. Těchto hodnot můžeme dosáhnout ovšem pouze v teoreticky dokonalých podmínkách, pokud je mezi komunikujícími zařízení překážka (např. zeď) dosah klesá. Nedochozí však ke ztrátě spojení, pouze ke zvyšujícímu se počtu chybných paketů.

Jako MAC adresa u Ethernetu, i Bluetooth zařízení jsou identifikovány jednoznačnou adresou *BD\_ADDR* (Bluetooth Device Address). Mezi Bluetooth zařízeními je možné vytvořit jak symetrický, tak asymetrický spoj, což znamená, že rychlosti příjmu jsou u symetrického spoje stejné a u asymetrického se liší, většinou vyšší rychlostí pro příjem dat. Technologií Bluetooth můžeme navázat spojení mezi dvěma a více zařízeními, pokud spojíme více zařízení, chová se jedno jako master a najednou dokáže obsloužit až 7 dalších zařízení. Podle specifikace můžeme mít na malé ploše až 10 těchto sítí.

## 2.3 Protokoly

Bluetooth používá řadu protokolů[7] pro komunikaci. Každý z těchto protokolů poskytuje určitou sadu možností komunikace. Nejzajímavější si shrneme v malém přehledu:

### L2CAP

L2CAP (*Logical Link Control and Adaptation Protocol*) poskytuje správu komunikace mezi jednotlivými Bluetooth zařízeními. Zapouzdřuje pakety do vhodného formátu a zajišťuje přenos tak, aby bylo možno komunikovat s více zařízeními. Základní vlastnosti L2CAP protokolu jsou:

- Maximální velikost packetu je 64KB.
- Zajišťuje jednotnou správu přenosu pro skupinu více bluetooth přístrojů.
- Poskytuje vyšším vrstvám spojové i bezspojové datové služby
- Hlídá kvalitu služeb - monitoruje kvalitu služeb dohodnutých pro spojení.

### RFCOMM

RFCOMM (*Radio Frequency Communication*) je jednoduchá sada protokolů založená na L2CAP protokolech, která emuluje RS-232 sériový port (až 60 současně připojených Bluetooth zařízení). Také je někdy nazýván Serial Port Emulation. Bluetooth seriové porty jsou na tomto protokolu založeny.

### BNEP

BNEP (*Bluetooth Network Emulation Protocol*) je používán v PAN profílech a slouží pro dopravu síťových paketů.

### OBEX

OBEX (*Object Exchange*) slouží k snadnější výměně binárních souborů mezi zařízeními.

## 2.4 Profily

Bluetooth v praxi používáme k mnoha různým činnostem, bezdrátové handsfree, poslech hudby, přenos souborů atd. Způsobů využití je opravdu mnoho a pro každou činnost je využíván jiný tzv. Bluetooth profil[6]. Profilů se vyskytuje celá řada, a každé zařízení podporuje jen určité z nich, u kterých se předpokládá, že budou v zařízení využity. Samozřejmostí je, že tyto profily musí být podporovány v obou komunikujících zařízeních.

### A2DP

A2DP (*Advanced Audio Distribution Profile*). Profil slouží k bezdrátovému poslechu hudby ve stereo kvalitě.

### AVRCP

AVRCP (*Audio/Video Remote Control Profile*) podporovaný zařízením dovoluje, již podle názvu, ovládat přehrávání hudby či videa (hlasitost, přeskokování mezi stopami, pozastavit nebo rozběhnout přehrávání atd.).

### DID

S profilem DID (*Device ID Profile*) snadně ideintifikuje právě připojené zařízení. Informace odesílané zařízením obsahuje údaje o jméně zařízení, výrobci a verzi zařízení.

### DUN

Jedním z nejběžněji používaných profilů je profil DUN (*Dial-up Networking Profile*), který umožňuje připojení zařízení (počítač, PDA) k internetu pomocí mobilního telefonu.

### FTP

Profil FTP (*File Transfer Profile*) poskytuje skrze bluetooth přístup k adresářům a složkám a práci s nimi.

### HFP

S profilem HFP (*Hands-Free Profile*) připojíte telefon k handsfree sadě a skrze ní telefonovat, přijímat či odmítat hovory atd.

### OPP

OPP (*Object Push Profile*) je základní profil určený pro posílání souborů, např. detailů schůzek a vizitek. Komunikace je vždy vyvolána a řízena odesílatelem, příjemce ji však může odmítnout.

### SDP

S pomocí SDP (*Service Discovery Profile*) vyšleme k námi určenému zařízení dotaz a obdržíme odpověď, jaké služby (resp. profily) podporuje.

## **SPP**

SPP (*Serial Port Profile*) umožňuje emulovat sériový port a řídit tak připojené zařízení. Je nezbytnou součástí pro fungování dalších profilů.

## **VDP**

VDP (*Video Distribution Profile*) jak již název napovídá, protokol umožňuje v reálném čase přenášet mezi zařízeními. Příkladem použití je streamování videa z počítače do mobilního telefonu, anebo přenos z kamery do počítače.

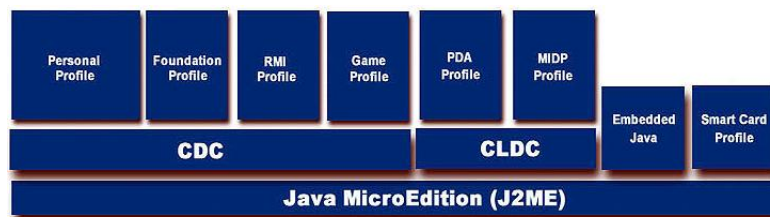
Výčet profilů je zde pouze zběžný a popisuje profily přímo spjaté pouze s mobilními telefony. Celkový výčet profilů a jejich popis je mnohonásobně delší.

# Kapitola 3

## J2ME

J2ME (*Java 2 Micro Edition*) [4] dnes nazývána Java Micro Edition (JME) je nejmenší z platforem Javy, určená pro malá zařízení s omezeným výkonem, jakými jsou například mobilní telefony, či kapesní počítače (PDA) a další jim podobná zařízení.

Mobilní Java je podmnožinou klasické Javy (J2SE). Sdílí s ní pouze určité části, a přidává své vlastní, čímž zajistí menší nároky na prostředky (paměť, procesor a síťové rozhraní), podle těchto nároků je také mobilní java rozčleněna, viz. obrázek 3.1. Vývoj J2ME zajišťuje proces JCP (*Java Community Process*). Každá konfigurace začala jako žádost JSR (*Java Specification Request*). Skupina odborníků poté vytvoří specifikaci a ta je poté schválena a poté se vytvoří návrh a zpřístupní se veřejnosti.



Obrázek 3.1: Základní členění J2ME.

### 3.1 Konfigurace

Konfigurace definuje požadavky pro určité typy malých zařízení a podle toho se dělí (např. klade minimální nároky na hardwarové vybavení zařízení). Každé z těchto zařízení má vlastní virtuální stroj, který se liší pouze v podporovaných funkcích podle konfigurace. V současné době se zařízení dělí do dvou skupin.

#### CDC

CDC (*Connected Device Configuration*) leží mezi CLDC a osobním počítačem. Jedná se o lépe vybavené zařízení s větším obsahem paměti (2MB a více) a výkonnějšími procesory. Mohou proto podporovat kompletní virtuální stroj jazyka Java. Často se tedy setkáme s touto konfigurací v dražších zařízeních PDA, chytrých telefonech, webových telefonech a dokonce i v zařízeních STB (Set-Top Box).

Zařízení CDC mají tyto minimální požadavky:

- 512 kB paměti ROM.
- 256 kB paměti RAM.
- Připojení k libovolnému typu sítě.
- Podpora kompletní implementace virtuálního stroje.
- Různě propracované uživatelské rozhraní.

## CLDC

S konfigurací CLDC (*Connected Limited Device Configuration*) se v souvislosti s J2ME<sup>[4]</sup> setkáme častěji. Je určena pro nejmenší zařízení. Typickým zařízením je mobilní telefon či PDA. Jedná se o nejmenší společný jmenovatel Javy.

Zařízení CLDC mají tyto požadované vlastnosti:

- 160 kB až 512 kB celkové paměti, minimálně 128 kB paměti pro virtuální stroj Javy a knihovny a 32 kB RAM paměti pro běh virtuálního stroje, vše nezávislé na jiných aplikacích.
- 16 nebo 32-bitový procesor s minimální taktovací frekvencí 25MHz.
- 2 a více barevný displej.
- Nízkou spotřebu energie, často napájení z baterie.

## 3.2 Virtuální stroje

Jak již bylo řečeno, pro každou z konfigurací existuje jiný virtuální stroj. Pro CLDC je to KVM (*Kilobyte Virtual Machine*) a pro konfiguraci CDC je to CVM (*Compact Virtual Machine*).

### KVM

KVM je virtuální stroj určený speciálně pro malé zařízení s omezenými zdroji, nicméně se jedná o opravdový virtuální dle specifikace pro virtuální stroje. Obsahuje odchylky, nutné pro funkčnost malých strojů. Je na každém z výrobců zařízení, jakým virtuálním strojem své zařízení vybaví, často se tedy stává, že se mohou virtuální stroje na různých zařízeních různých výrobců mírně lišit.

### CVM

CVM je virtuální stroj určený pro větší zařízení konfigurace CDC a podporuje všechny rysy JVM (*Java Virtual Machine*).

### 3.3 Profily

Konfigurace CLDC i CDC poskytují pouze základ pro běh Javy na malých zařízeních a profily slouží k doplnění takových prostředků tak, aby bylo možno se zařízením pracovat. Každý profil je navržen pro určitý způsob využití a k tomuto využití nabízí vhodné prostředky. Rozdělení profilů pro konfigurace je vidět na obrázku 3.1.

#### PDA

Profil PDA jak už název napovídá je určen pro výkonnější zařízení s větším displejem, více pamětí a výkonnějšími procesory.

#### Základový profil

Základový profil (*Foundation Profile*) rozšiřuje CDC o téměř všechny standardní knihovny, které obsahuje jádro Javy 1.3 a slouží jako základ pro ostatní profily CDC.

#### Osobní profil

Osobní profil (*Personal Profile*) přidává funkce uživatelského rozhraní k profilu základovému. Jedná se o prostředí s plnou podporou AWT (*Abstract Window Toolkit*).

#### RMI profil

RMI (*Remote Method Invocation*) profil jak už název napovídá přidává k základovému profilu knihovny pro vzdálené vyvolávání metod J2SE.

#### MIDP

MIDP (*Mobile Information Device Profile*) je založen na CLDC a KVM a nabízí třídy pro práci v síti, práci s uživatelským rozhraním a trvalé ukládání dat. Jedná se o nejznámější profil J2ME a tvoří základ bezdrátové Javy (Wireless Java). Aplikace této kategorie se nazývá *Midlet*. Hardwarové nároky přebírá od specifikace CLDC a dále je rozšiřuje:

- Minimálně 2 barevný displej a velikost 96 x 54 pixelů.
- Klávesnice nebo dotyková obrazovka.
- Minimálně 128 KB stálé paměti pro komponenty.
- Minimálně 32 KB paměti pro Javu.
- Minimálně 8 KB pro data aplikací.
- Připojení k síti.

#### Třídy profilu MIDP

Pro vývoj aplikace si povíme pouze o některých třídách profilu MIDP. Pro vytvoření prvního Midletu musíme naimplementovat třídu, která bude rozšiřovat abstraktní třídu *MIDlet* a musí obsahovat:

- Bezparametrický konstruktor.

- Metodu `startApp()`.
- Metodu `pauseApp()`.
- Metodu `destroyApp(boolean unconditional)`.

Tyto metody jsou povinně vyžadovány specifikací třídy *Midlet* pro správný běh midletu, již názvy metod napovídají kdy a jak jsou vyvolávány, implementace těl midletů však nejsou povinné.

Pokud uživatel požaduje, aby aplikace reagovala na je pokyny, je nutné vytvořit objekty typu *Command* a třída Midletu musí implementovat *CommandListener*, poté již ve třídě midletu musíme naimplementovat metodu `CommandAction(Command c, Displayable d)`, kde můžeme těmto příkazům definovat jednotlivé akce, které se po vyvolání příkazu budou provádět.

Pro získání přístupu k displeji mobilního slouží metoda třídy *Display*, `getDisplay(MIDlet midlet)`, která vrací objekt typu *Display* požadovaného midletu. Tomuto objektu poté metodou `setCurrent(Displayable d)` můžeme přiřadit objekt, který má být aktuálně zobrazen.

Na displeji bývají nejčastěji zobrazovány objekty typu *Form*, který představují formulář, a je na něj možné umístit jednoduché komponenty a nebo objekt třídy *Canvas*, který nabízí více možností interakce s uživatelem. Objekt třídy *Canvas* vyžaduje mít implementovanou vlastní třídu, protože třída *Canvas* je pouze abstraktní. V této třídě jsou povinné dvě metody `paint(Graphics g)`, která je vyvolána vždy, pokud chceme zobrazitelnou plochu překreslit a metoda `keyPressed(int keyCode)`, která je sama vyvolána při stisku klávesy mobilního zařízení.

Kompletní výčet tříd a metod by sám vydal na publikaci a je možné jej nalézt zde<sup>[4]</sup>.

## MIDP 2.0

Když se v mobilních telefonech objevila Java, způsobila malou revoluci, když uživateli umožnila, aby si sám volil, jaké programy chce mít ve svém přístroji uloženy. S postupem času se vybavení mobilních telefonů vyvíjelo a bylo nutné vyvinout i mocnější nástroj pro práci s nimi. MIDP 2.0 předpokládá, že zařízení obsahuje CLDC verzi 1.1 a mobilní telefon podporuje kvůli zpětné kompatibilitě i profil MIDP verze 1.0. MIDP 2.0 zvětšila svůj objem tříd, a proto zvýšila i své nároky na hardware. Vyžaduje 256 kB RAM paměti a 128 kB paměti pro běh JRE a také zabraňuje kopírování celých sad midletů, čímž by měla zabránit krádežím programů. Největším vylepšením oproti MIDP 1.0 je práce s grafikou, MIDP 2.0 dovoluje tvořit složitější grafická rozhraní.

Dnes již na každém telefonu najdeme MIDP verze 2.0.

## 3.4 J2ME a Bluetooth

V mobilní Javě samozřejmě existuje podpora pro bezdrátovou komunikaci pomocí Bluetooth. Toto rozhraní je definováno v čísle požadavku JSR 82, ve kterém je implementována práce s několika profily Bluetooth (L2CAP, RFCOMM, OBEX). Jak s rozhraním v Javě pracovat lze nalézt zde<sup>[3]</sup>.

## Kapitola 4

# Neformální specifikace

Cílem projektu je osvobození přednášejícího od neustálé přítomnosti u klávesnice při ovládní prezentace. Nástroje, které podobné možnosti nabízejí, již na trhu dostupné jsou, ale často jsou jejich pořizovací náklady zbytečně vysoké a jejich jiné využití je prakticky nulové. Naším cílem je zpřístupnit tuto možnost většímu počtu uživatelů, kteří si nemohou, nebo prostě nechtějí toto řešení pořizovat. Nejprve si ale musíme upřesnit, co vlastně od tohoto řešení očekáváme.

### 4.1 Požadavky na hardware

#### Mobilní telefon

Ke komunikaci je nutné vlastnit mobilní telefon s dvou a vícebarevným displejem a podporou Bluetooth pro komunikaci s počítačem a přiměřenou paměťovou rezervou pro instalaci, uchovávání a běh aplikace.

#### Počítač

Počítač musí být vybaven operačním systémem Windows XP, prezentačním programem, který reaguje na vstup z klávesnice alespoň pro přechod mezi snímky. Bluetooth modulem, který bude zajišťovat komunikaci s mobilním telefonem, ať už ve formě vestavěného nebo přídatného modulu.

### 4.2 Požadavky na aplikaci

Aplikace musí být navržena tak, aby i obyčejný uživatel byl schopný aplikaci s pomocí přiloženého manuálu nainstalovat, spustit a ovládat ať už v počítači, nebo v mobilním telefonu.

### 4.3 Požadavky na funkčnost

#### Počítač

Uživatel spáruje mobilní telefon s počítačem (povolí komunikaci) a spustí jednoduchou aplikaci, která bude vyčkávat na připojení mobilního telefonu. Po navázání spojení bude vykonávat příkazy přijaté od připojeného mobilního telefonu.



## **Mobilní telefon**

V mobilním telefonu uživatel zapne funkci Bluetooth a spustí Java aplikaci s jednoduchým GUI, pomocí které bude moci ovládat prezentaci. Pokud uživatel sám Bluetooth nespustí, aplikace se jej na spuštění Bluetooth dotáže. Poté bude uživatel moci ovládat vzdáleně prezentaci pomocí výběru z nabídky Menu nebo stiskem numerických kláves.

Hlavním cílem je co nejširší využití na vysokém počtu mobilních telefonů různých výrobců, omezené pouze podporou technologie Bluetooth a mobilní Javy.

# Kapitola 5

## Analýza

### 5.1 Bezdrátová komunikace

Dnešní mobilní telefony i počítače jsou vybaveny několika možnostmi bezdrátové komunikace, představíme si ty nejznámější.

#### IrDA

IrDA (*Infrared Data Association*) definuje standardy protokolů používané při komunikaci mezi zařízeními pomocí infračerveného modulovaného světla o vlnové délce 875 nm. Vzhledem ovšem k malému dosahu a zadání projektu se dále nebudeme touto technologií zabývat.

#### Wi-Fi

Je standardem definujícím lokální bezdrátové sítě (Wireless LAN). Wi-Fi sítě jsou v dnešní době mezi počítači velice rozšířené a dostávají se i do lépe vybavených telefonů, bohužel tato technologie není stále obsažena v dostatečně velkém množství mobilních telefonů, aby vyhovovala zadání, kde je kladen důraz na co nejširší použití.

#### Bluetooth

Technologie Bluetooth je již popsána v druhé kapitole a vzhledem ke znění zadání je proto pro nás jedinou technologií komunikace mezi telefonem a počítačem. Technologie Bluetooth je dostupná v několika specifikacích (výkonová třída, verze atd.), které ovlivňují faktory jako dosah či přenosovou rychlost. Tyto specifikace jsou již součástí zařízení (mobilní telefon, PDA aj.) a vývojáři aplikací je nemohou ovlivnit. Dnešní mobilní zařízení ovšem téměř ve většině případů podporují Bluetooth verze 2.0 s přenosovou rychlostí 3Mb/s a minimálně třídy 2 (dosah 10 metrů), které našim požadavkům vyhovují. Přenosová rychlost by mohla být i menší, protože nebudeme přenášet velké soubory, a dosah 10 metrů je pro přednášejícího dostačující.

### 5.2 Softwarové vybavení telefonu

#### Symbian

Symbian je operačním systémem, který byl navržen speciálně pro tzv. chytré telefony. Je vyvíjen firmou Symbian a je vydáván v několika verzích. Vzhledem k jeho malé rozšířenosti

v mobilních telefonech (např. chytré telefony Nokia) musíme hledat takový nástroj, který je podporován větším množstvím mobilních telefonů.

## Windows Mobile

Windows Mobile jsou operačním systémem vytvořeným firmou Microsoft a založeným na Win32 API z Microsoft Windows. Rozdíl mezi Symbianem a WM je v podpoře dotykových displejů. Verze Symbianu UIQ ovšem spojuje oba OS a vytváří spojnicu mezi nimi. Pro projekt jsou ovšem nepodstatné. Stále jsou doménou pouze tzv. „Chytrých telefonů“.

## Java Micro Editon

J2ME je rozšířena téměř ve všech mobilních telefonech a je námi hledaným společným nástrojem pro tvorbu aplikace, a proto je jí věnována celá třetí kapitola. Podporou mobilní Javy je dnes vybaven téměř každý mobilní telefon, a proto je vyhledávána tvůrci aplikací (hry a jiné) pro mobilní telefony, a i v naší práci ji zvolíme jako vývojový nástroj. Pro mobilní telefony je určena konfigurace CLDC, z které vychází profil MIDP verze 2.0 a i tuto specifikaci použijeme v naší aplikaci.

## 5.3 Programovací jazyky a Bluetooth

Podporu komunikace mezi zařízeními Bluetooth nabízí několik programovacích jazyků např. z rodiny jazyků C (C++, C#), které používají přímo knihovny k API operačního systému Windows. My ovšem zvolíme jazyk Java, v kterém bude napsána klientská část, a pro práci s Bluetooth využijeme knihovnu BlueCove [5], která nabízí stejné funkce jako JSR 82 pro J2ME, čímž bude programování klienta i serveru velice podobné a zajistí bezproblémovou komunikaci.

## 5.4 Prezentační programy

Už i výklady na základní škole byly podporovány například obrázky nebo ukázkami z map a podobnými demonstračními nástroji.

Nyní již mají přednášející k dispozici techniku, která dokáže nahradit mapy a obrázky pouze monitorem a vhodným prezentačním programem. Mezi uživateli jsou oblíbené prezentační nástroje skládající se z několika různých snímků, na kterých se prezentující snaží podpořit či doplnit svůj výklad osnovou, obrázky, či hlavními body prezentace.

Ovládání každého prezentačního programu vyžaduje přítomnost u klávesnice, či jiného vstupního zařízení. Ovládat počítače myšlenkou je zatím stále záležitost spíše science fiction.

Podíváme se na několik příkladů prezentačních programů a jejich ovládání pro některé úkony zobrazené v tabulce 5.1.

| Název programu       | Další | Předchozí | První | Poslední | Start    | Konec    |
|----------------------|-------|-----------|-------|----------|----------|----------|
| Adobe Reader         | Right | Left      | Home  | End      | Ctrl + L | Ctrl + L |
| Microsoft PowerPoint | Right | Left      | Home  | End      | F5       | Esc      |
| OpenOffice Impress   | Right | Left      | Home  | End      | F5       | F5       |

Tabulka 5.1: Ovládací prvky tří nejužívanějších prezentačních programů.

Jak vidíme z tabulky 5.1 řada příkazů se opakuje, ale některé příkazy se mohou lišit, proto by bylo vhodné zvolit jako rozšíření práce, aby si uživatel sám mohl tyto příkazy definovat.

## 5.5 Existující zařízení

### Jednouúčelové ovládací zařízení

Osvobodit přednášejícího od neustálé přítomnosti u klávesnice či myši při přednášení pomáhají již dostupná prezentační zařízení od několika společností poskytující řešení pro ovládání prezentace bezdrátově. Nejčastější poskytované funkce jsou kromě přechodů mezi snímky, i pohyb kurzorem, či laserové ukazovátko. Jejich zpracování, výrobci a bohatost funkcí se liší, co se ovšem neliší, je cena, pohybující se kolem jednoho tisíce korun. Jedno z takových zařízení můžeme vidět na obrázku 5.1.



Obrázek 5.1: Prezentační zařízení od firmy Dicota.

### Softwarové řešení

Pro kapesní počítače existuje podobné řešení, jako např. tato práce [2], z které jsem při své práci částečně vycházel. Práce je ovšem omezená pouze na základní ovládací prvky a k jejímu spuštění je nutný kapesní počítač.

# Kapitola 6

## Návrh

Řešení bude sestaveno ze dvou částí. Části pro mobilní telefon (klient) a z části pro počítač (server). Stačí aby komunikace probíhala pouze jedním směrem klient - server, obrácená komunikace je pro naši aplikaci zbytečná. Klient i server spolu musí mít předem domluvenou sadu zpráv. Výčet zpráv zasílaných mobilním telefonem serveru je následovný:

- Další/předchozí snímek.
- První/poslední snímek.
- Spustit/ukončit prezentaci.
- Snížit/zvýšit hlasitost.
- Vypnout/zapnout zvuk.

Klientská část bude dle analýzy vytvořená v jazyce J2ME v konfiguraci CLDC a profilu MIDP 2.0 využívající rozhraní Javy pro Bluetooth JSR 82.

Serverová část bude dle analýzy vytvořená v jazyce Java využívající knihovny BlueCove[5]. Samozřejmostí je předchozí spárování mobilního telefonu s počítačem, aby komunikace mohla probíhat bez problému.

### 6.1 Mobilní telefon

#### Spuštění

Po spuštění klientské části aplikace bude automaticky spuštěno vyhledávání ostatních Bluetooth zařízení v okolí a poté bude nabídnut výběr jednoho z nich, ke kterému se opravdu budeme chtít připojit a komunikovat s ním. Poté co proběhne navázání komunikace již budeme očekávat pouze příkazy od uživatele, které se mají zaslat serveru, aby je vykonal.

#### Uživatelské rozhraní

Uživatelské rozhraní je důležité pro celkovou pohodlnost ovládání aplikace, při jeho tvoření je nutné dbát na univerzálnost celé aplikace, proto se při návrhu rozhraní omezíme na jednoduché ovládání prvků tak, aby byly vždy po ruce a jejich vzhled byl stejný na co největším počtu mobilních zařízení, aby uživatele nemátlo různé zobrazení.

Pro nejpoužívanější příkazy, přechod na další či předchozí snímek, jsem zvolil stisk jedné z řady kláves pod sebou (1,4,7 nebo 3,6,9). Po spuštění aplikace budou klávesy 3, 6 a 9 reprezentovat přechod na další snímek a klávesy 1, 4 a 7 přechod na předcházející snímek. Samozřejmě musí být prohození funkcí obou skupin kláves pro uživatele, který by to vyžadoval (pohodlnější obrácené držení telefonu a podobně). Pro další funkce, které uživatel již nebude užívat tak často, bude možnost vybrat si z jednoduchého Menu, kde by položky mohly vypadat a být rozloženy jako na obrázku 6.1. Na obrázku je také vidět, jak jsou klávesy 1, 4, 7, 3, 6 a 9 rozloženy na mobilním telefonu klasické konstrukce.



Obrázek 6.1: Rozložení ovládacích prvků klientské části aplikace.

## Programová část

Program vytvořený pro mobilní telefon se bude skládat ze samostatných, spolupracujících oddílů, kde každý oddíl bude reprezentovaný třídou jazyka Java, obsluhující jednu určitou část aplikace. Tyto třídy by mohly být následující:

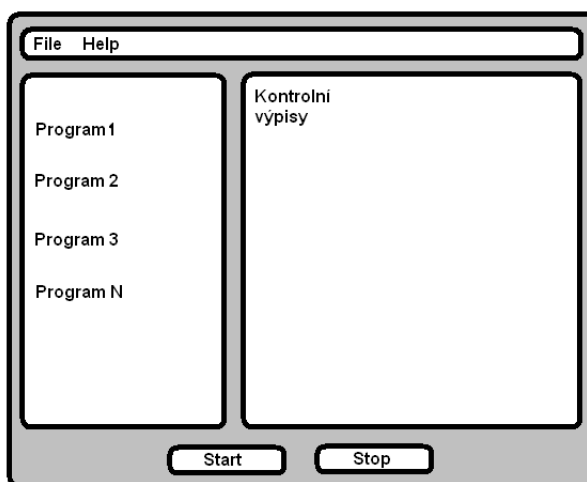
- **BFClient**  
Tato třída se sama bude starat o navázání spojení se serverem, vyhledání služby na serveru a zaslání zpráv serveru.
- **BFMidlet**  
Tato třída bude první spuštěnou třídou na mobilním telefonu, vyvolá start klientské části, obstará vybrání serveru a bude uchovávat informace o běhu programu pro jejich možnou zpětnou kontrolu.
- **BFCanvas**  
Tato třída bude obsluhovat zpracování příkazů zadaných uživatelem a také zajistí ob-

sluhu jednoduchého grafického výstupu informujícího uživatele o aktuálním nastavení ovládacích prvků.

## 6.2 Počítač

### Spuštění programu

Po spuštění programu se uživateli nabídne pouze jednoduché okno, které by mohlo vypadat jako na obrázku 6.2. Bude pouze na uživateli, jaký prezentační program z nabídky zvolí. Dále spustí běh serveru, který bude přijímat zprávy od klienta a vykonávat jeho příkazy.



Obrázek 6.2: Okno aplikace.

### Uživatelské rozhraní

Jediné uživatelské rozhraní, s kterým přijde uživatel do styku je okno, které uvidí uživatel hned po spuštění. V jeho horní části najde uživatel menu, které nabídne rozšiřující možnosti kontrolních výpisů do pravé části *Kontrolní výpisy*, díky tomuto výpisovému oknu bude mít uživatel přehled o provedených nebo právě prováděných akcích (start serveru, přijatá zpráva od klienta atd.) a díky tomuto oknu si bude uživatel moci zkontrolovat, zda server dělá to co má a co uživatel vyžaduje. V levé části najdeme výběr prezentačních programů z kterých si uživatel bude moci sám vybrat a podle tohoto výběru bude server provádět úkony. Dole v okně najdeme tlačítko pro start a konec běhu serveru.

### Rozšíření funkčnosti

Jako možné rozšíření nad rámec zadání umožníme uživateli, aby si sám nadefinoval pro ovládání prezentace svoje klávesové zkratky. Příkazová část programu bude fungovat na základě simulace stisku kláves klávesnice. Uživateli umožníme editaci jednoduchého XML souboru jejich vlastní definování.

## Programová část

Program bude napsán v jazyce Java využívající knihovnu BlueCove pro komunikaci pomocí Bluetooth a bude se skládat z těchto vzájemně komunikujících částí reprezentovaných třídami:

- **BluetoothServer**  
Tato třída se sama bude starat o start serveru, navázání komunikace a příjem a vykonávání příkazů od klienta.
- **BlueFinger**  
Tato třída se postará o spuštění zobrazitelného okna, načtení příkazů z XML souboru.
- **ServerFrame**  
Třída se bude starat o vzhled samotného okna, bude obsluhovat události reagující na určité akce.
- **XmlHandler**  
Třída, která se bude starat o zpracování XML dokumentu, načítání z něj a podle vstupu z XML souboru, budou vyvolávány akce.

## Protokol a klávesové zkratky

Komunikační protokol bude založený na zasílání celých slov od klienta k serveru. Slova jsou složena z ASCII znaků zakončených značkou \$\$ pro zajištění korektnosti zaslaných a přijatých dat. Tabulka 6.1 ukazuje, jak vypadají jednotlivé příkazy pro určité úkony.

V XML souboru, který bude definovat pro jednotlivé příkazy úkony stisknuté klávesnicí budou již uvedené klávesové zkratky pro jednotlivé klávesy definované v příslušné dokumentaci jazyka Java [8]. V XML souboru bude možná definice stisku několika kláves současně, toto je kvůli možným klávesovým zkratkám (Ctrl + L apod.).

| Příkaz                | Zpráva |
|-----------------------|--------|
| Další snímek          | NEXT   |
| Předchozí snímek      | BACK   |
| První snímek          | HOME   |
| Poslení snímek        | END    |
| Start prezentace      | FULL   |
| Konec prezentace      | STOP   |
| Vypnutí/zapnutí zvuku | VMUTE  |
| Zesílení zvuku        | VUP    |
| Zeslabení zvuku       | VDOWN  |
| Ukončení komunikace   | BYE    |

Tabulka 6.1: Komunikační protokol.



## Kapitola 7

# Implementace

### 7.1 Komunikace pomocí Bluetooth

Tato část zdála se být a byla stěžejní v celém projektu, navázat spojení mezi mobilním telefonem s počítačem představovalo první velkou výzvu. Nejprve jsem se rozhodl vyzkoušet si navázat spojení mezi dvěma mobilními telefony podle srozumitelného návodu [3] po delší době a malých či větších obtížích se mi podařilo vytvořit aplikaci, která na jednom mobilním telefonu spustí server a na druhém klienta a zašle zprávu serveru. Vytvoření spojení demonstruje následující obrázek 7.1 převzatý z [3].

Jak je z obrázku patrné, serverová část je podstatně jednodušší než část klientská. Na serveru pouze vytvoříme a spustíme službu a už jen čekáme na příchozí spojení. Klientská část je o něco složitější, po inicializaci Bluetooth adaptéru v zařízení, je nutné nejprve vyhledat všechna dostupná zařízení v okolí a pak v každém postupně, či pouze v jednom zvoleném vyhledat požadovanou službu, ke které se poté chceme připojit. Pokud ani na jednom zařízení není dostupná adekvátní služba, komunikace probíhat nebude.

Při navazování spojení s počítačem jsem ovšem narazil na problém, klasická Java nepodporuje mobilní JSR 82 a bylo proto hledat adekvátní a podobné řešení, které jsme našli v opensourcové knihovně BlueCove [5], což je vlastně implementace JSR 82 pro klasickou Javu pod operační systém Microsoft Windows. Pokud bychom chtěli hledat toto řešení pro operační systémy GNU Linux, knihovna se jmenuje BlueZ. S importováním této knihovny mi již nic nebránilo v komunikaci mezi počítačem a mobilním telefonem.

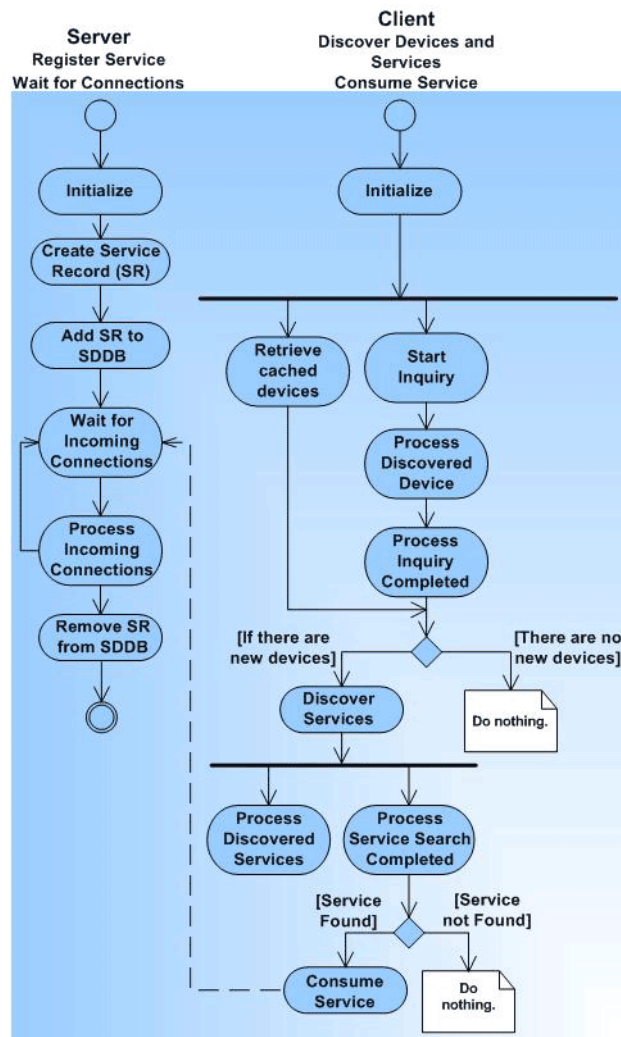
### 7.2 Serverová část

Jak již bylo zmíněno dříve, serverová část je celá napsaná v jazyce Java skládající se z několika základních částí:

- Komunikačního serveru a správce příkazů.
- Správce grafického rozhraní a příkazů.

a rozšiřujících částí:

- Zpracování XML souboru.
- Správce hlasitosti.



Obrázek 7.1: Schéma komunikace.

## Komunikační server

Komunikační server má za úkol, s využitím knihovny BlueCove vytvořit serverovou část spojení. Dále vytvořit službu, přijímat, zpracovávat a vykonávat přijaté požadavky. Na problém jsem narazil při vykonávání požadavků klienta. Pro simulaci stisku kláves klávesnice byly použity metody z třídy `Robot` z balíku `java.awt` a jejich metody `KeyPress(int keycode)` a `KeyRelease(int keycode)`, kde `keycode` je např. `VK_UP`, více zde[8].

Samotný komunikační server je zastoupen třídou `BluetoothServer` která má následující důležité metody:

- `BluetoothServer(ServerFrame sf)`  
 Jedná se o konstruktor třídy, vstupem kterého je objekt třídy `ServerFrame`, což je třída zpracovávající grafické rozhraní. Je předáván proto, aby v této třídě bylo možné provádět výpisy do okna grafického rozhraní.

- `btInit()`  
Tato třída provede prvotní inicializaci serveru, získání lokálního zařízení a nastavení viditelnosti zařízení na neomezenou dobu.
- `run()`  
Kvůli možnosti zpracovávání příkazů uživatele z grafického rozhraní a souběžného běhu serveru je pro běh serveru vytvořeno nové vlákno, které se spouští metodou `run()`, tato metoda v sobě nejdříve vyvolá metodu `startServer()`, která nastaví serveru unikátní UUID a otevře spojení a čeká na připojení klienta. Po připojení klienta je akceptováno spojení a otevřen proud pro příjem zpráv.
- `PushButton(String s)`  
Metoda, která je zavolána po příjmu dat od klienta a přebírá řetězec na vstupu, podle kterého vykoná příslušnou akci.
- `stopServer()`  
Metoda, která zastaví běh serveru.

## Správce grafického rozhraní

Správce grafického rozhraní (třída `ServerFrame`) je třída rozšiřující třídu `JFrame`, která se stará o správné rozložení prvků a zpracování událostí vyvolaných komponentami třídy. Každá z částí je vytvořena pro určitý úkol. Prvky `RadioButton` v levé části okna reprezentují výběr klávesových zkratk z XML dokumentu. Po změně výběru aktuálního prvku je automaticky vyvolána metoda `actionPerformed(ActionEvent e)`, která si uvnitř sebe vyvolá metodu pro načtení dat z dokumentu třídy `XmlHandler`, která poskytuje metody pro práci s XML dokumentem.

V pravé části okna je `Output panel`, kam jsou zapisovány výstupní informace místo klasického konzolového výstupu. Pro uživatele se jedná o kontrolu o prováděných akcích a stavu běhu aplikace.

Pod tímto jsou dvě tlačítka, `Start` a `Stop`, která spouští, či zastavují běh vlákna serveru.

## Zpracování XML souboru

Jako rozšíření aplikace jsem zvolil uživatelem editovatelný soubor `control.xml`, který mu umožňuje definovat vlastní klávesové zkratky pro další možné prezentační programy, které se dají ovládat z klávesnice. Třída pro zpracování XML souboru má název `XmlHandler` a poskytuje metody, které dokáží přistupovat k jednotlivým prvkům souboru pro všechny úrovně zanoření.

Knihoven pro práci s XML dokumenty je celá řada, nicméně já jsem zvolil mnou osvědčenou knihovnu `SimpleXML`, která nabízí zcela jednoduchou práci s XML soubory. Více o této knihovně lze najít zde [9].

Struktura XML souboru je následující. Prvek `<root>` je hlavním prvkem ohraničujícím celý dokument. Prvek `<program>` s argumenty `name`, který označuje název prezentačního programu, a `id`, určujícím identifikátor, ohraničuje typ prezentačního programu a prvek `action` s argumentem `name` určuje akci a prvek `key` s argumentem `code` určuje kód akce definovaný zde [8].

Prvků `program` může být libovolný počet, prvků `action` by mělo být 6. Prvků `key` může být libovolný počet, nejméně však jeden. Ukázka XML struktury je v příloze C.

Každý prvek XML souboru je reprezentován svou vlastní třídou, což je dáno prací s knihovnou SimpleXML.

## Správce hlasitosti

Správce hlasitosti je další rozšíření aplikace a je reprezentován třídou `Controler`, která využívá tříd a metod z balíku `javax.sound`. Tato třída obsahuje metody:

- `getSpeaker()`  
Tato metoda získá port, na kterém je k dispozici speaker.
- `getSpeakerVolLevel()`  
Metoda vrací úroveň hlasitosti speakru.
- `setSpeakerVolLevel()`  
Metoda nastaví úroveň hlasitosti na speakru.

## 7.3 Klientská část

Klientská část je implementována v jazyce J2ME profilu MIDP 2.0 a skládá se ze dvou částí:

- Komunikační část.
- Část grafického rozhraní.

Při implementaci této části, bylo největším problémem zkloubit ovládání a zobrazení, aby se chovalo stejně na různých mobilních telefonech.

### Komunikační část klienta

Komunikační část klienta je reprezentována třídou `BFCClient` která postupuje podle návodu[3]. Třída obsahuje tyto metody:

- `run(BTMidlet midlet)`  
Je prvotní metoda spuštění klienta, která vytváří spojení na objekt třídy `BTMidlet` pro získání sady pro výpis na kontrolní výstup.
- `deviceDiscovered(RemoteDevice btDevice, DeviceClass cod)`  
Tato metoda je první ze čtyř povinných metod pro práci s Bluetooth, vyhledává vzdálené zařízení.
- `inquiryCompleted(int discType)`  
Je další z povinných metod a je zavolána bezprostředně poté, co cyklus vyhledávání vzdálených zařízení skončil.
- `servicesSearch (Vector listOfDevice)`  
Hledá v seznamu zařízení služby na uživatelem zadaném serveru, pro vyhledání určité služby používá metodu `serviceSearch()`.
- `servicesDiscovered(int transID, ServiceRecord[] records)`  
Tato metoda je další povinnou metodou, která je vyvolána po skončení prohledávání služeb.

- `serviceSearchCompleted(int transID, int respCode)`  
Poslední povinnou metodou, která otevírá spojení společně s komunikačním kanálem kam se bude zapisovat.
- `sendMessages(String strData)`  
Metoda, která na výstupní kanál zapíše řetězec znaků `strData`.
- `closeClient ()`  
Bezpečně uzavře spojení mezi klientem a serverem.

## Část komunikačního rozhraní

Tato část je reprezentována dvěma třídami `BFMidlet` a `BFCanvas`.

První z nich povinně rozšiřuje třídu `MIDlet` a implementuje třídu `CommandListener`, která nabízí uživateli tři povinné metody :

- Konstruktor třídy, který získá sadu nástrojů displeje, vytvoří nový formulář a přidá tlačítka na displej.
- `startApp()`  
Tato metoda je povinná pro třídu `MIDlet` a je vyvolána po startu aplikace a spustí běh klienta, vytvoří nový formulář a zobrazí jej na displeji. Je také vyvolána po minimalizaci aplikace, a proto musíme hlídat spuštění klienta pouze jednou.
- `pauseApp()` a `destroyApp()`  
Tyto dvě metody jsou povinné jako metoda `startApp()`, jejich tělo je ovšem prázdné.
- `comandAction (Command command, Displayable d)`  
Metoda je povinná pro `CommandListener`, reaguje na příkazy uživatele.
- `logMessage(String s)`  
Zajišťuje vypsání kontrolních zpráv na formulář.
- `choiceServerName(Vector listOfDevice)`  
Dovoluje uživateli vybrat ze seznamu zařízení nalezených v okolí to, v kterém má hledat službu pro komunikaci s klientem.

Další třída `BFCanvas` rozšiřuje abstraktní třídu `Canvas`, která umožňuje složitější grafický výstup a odchyťávání zpráv stisku, držení či uvolnění tlačítek a dotykové obrazovky mobilního telefonu.

- `paint(Graphics g)`  
Je povinná metoda sloužící k překreslení obsahu displeje. Je vyvolána na základě volání metody `repaint()`. Okno je rozděleno na dvě části, které se vykreslují samostatně, barva vykresleného podkladu záleží na tom, který příkaz (`Next`, `Back`) byl požadován a ta strana okna je vyznačena oranžovou barvou.
- `keyPressed(int keyCode)`  
Metoda je zavolána vždy, při stisku tlačítka uživatelem, a v jejím těle je volána metoda pro vykonání příkazu.
- `pointerPressed(int x, int y)`  
Reaguje na stisk bodu na dotykovém displeji.

- `doCommand (int keyCode)`  
Rozhoduje na základě vstupní hodnoty jaký příkaz má být odeslán a zohledňuje prohození stran ovládání.
- `commandAction(Command command, Displayable d)`  
Na základě vybrané položky z menu obstará odeslání příkazu serveru či vyvolání správné požadované akce.

# Kapitola 8

## Závěr

### 8.1 Shrnutí

Cílem práce bylo osvobodit prezentujícího od neustálé přítomnosti u počítače při přechodu mezi snímky. Na trhu jsou již dostupná řešení v podobě dálkových prezentačních ovladačů, jejich cena je ovšem zbytečně vysoká. Dalším možným řešením je použít podobné aplikace, kde žádná z nich nenabízí stejné funkce jako ta naše, včetně rozšiřujícího ovládání hlasitosti a vlastní definice příkazů pomocí XML souboru.

Vytvořili jsme tedy jednosměrně komunikující řešení, které mohou využít všichni běžní uživatelé. Aplikaci si navíc mohou uživatelé přizpůsobit tak, aby ovládala jakýkoliv program řízený klávesnicí, omezení jsou pouze počtem definovatelných ovládacích příkazů.

Během vytváření všech součástí programu jsem se zaměřil na to, aby využití aplikace bylo co nejširší a bylo dostupné opravdu každému. Během implementace byla aplikace testována na několika různých mobilních telefonech, především značek Nokia a Sony Ericsson. Všude fungovala bez nejmenší problémů. U Serverové části jsem ovšem narazil na omezenost knihovny BlueCove, která nedokázala přistupovat k Bluetooth modulům firmy Toshiba, nicméně vývojáři této knihovny o problému ví a chystá se jej odstranit.

Celá práce je k nahlédnutí na přiloženém CD, jeho obsah lze nalézt v příloze A a návod jak aplikaci ovládat v příloze B.

### 8.2 Budoucí vývoj

Tato práce poskytuje dostatečné možnosti k pohodlnému ovládání desktopové prezentace, ale stále se nabízí velké množství možností, jak ji obohatit:

- Kompletní podpora dnes již značně rozšířené dotykové obrazovky, pro kterou by se muselo upravit grafické rozhraní klientské části aplikace. Některým uživatelům nemusí vyhovovat stisk klávesy, popř. některé mobilní telefony nejsou hardwarovou klávesnicí vybaveny, a proto by bylo lepší ovládat aplikaci pouze dotykovým displejem.
- Rozšíření příkazů a komunikačního protokolu tak, aby se ovládání neomezovalo pouze na ovládání prezentací, ale na ovládání dalších funkcí počítače. Například multimediální programy a další. Toto rozšíření je již teď částečně možné, pokud by si uživatel sám přidělil určitým příkazům jiné úkony.

- Rozšíření komunikace mezi zařízení pomocí dalších technologií (Wi-Fi, GPRS, IrDA aj.). Toto řešení by mělo smysl, pokud bychom chtěli např. v případě Wi-Fi posílat i soubory či obsáhlé zprávy.
- Podpora poznámek. Uživateli by se mohla ke každému ze snímků do mobilního telefonu zobrazit krátká poznámka k prezentaci. K tomuto by bylo nutné navázat obousměrnou komunikaci, kvůli provázanosti poznámek s aktuálním snímkem.

Možnosti rozšíření aplikace jsou rozsáhlé a jejich realizace by vydala na více prací.



# Literatura

- [1] BAKKER, D. M.; GILSTER, D. M.; GILSTER, R.: *Bluetooth End to End*. New York, NY, USA: John Wiley and sons Inc., první vydání, 2002, ISBN 978-0-7645-4887-1.
- [2] LIŠKA, J.: *Bezdrátové ovládání prezentací pomocí kapesního počítače*. [bakalářská práce], Vysoké učení technické v Brně. Fakulta informačních technologií, 2007, vedoucí práce Ing. Petr Novosad.
- [3] ORTIZ, C. E.: Using the Java APIs for Bluetooth. [online], 2005, [cit. 1. května 2009]. URL <http://developers.sun.com/mobility/apis/articles/bluetoothcore/>
- [4] TOPLEY, K.: *J2ME v kostce - Pohotová referenční příručka*. Praha: Grada, O'Reilly, první vydání, 2004, ISBN 80-247-0426-9.
- [5] Bluecove JSR-82 project. [online], 2004 - 2008, [cit. 1.května 2009]. URL <http://www.bluecove.org/>
- [6] Bluetooth profily. [online], [cit. 1.května 2009]. URL [http://en.wikipedia.org/wiki/Bluetooth\\_profile](http://en.wikipedia.org/wiki/Bluetooth_profile)
- [7] Bluetooth protokoly. [online], [cit. 1.května 2009]. URL [http://en.wikipedia.org/wiki/Bluetooth\\_protocols](http://en.wikipedia.org/wiki/Bluetooth_protocols)
- [8] KeyEvent Documentation. [online], 2003, [cit. 1. května 2009]. URL <http://java.sun.com/j2se/1.4.2/docs/api/java/awt/event/KeyEvent.html>
- [9] SimpleXML. [online], [cit. 1.května 2009]. URL <http://simple.sourceforge.net/>

# Příloha A

## Obsah CD

Obsah přiloženého CD je následující:

- soubor *bp.pdf*  
je textová podoba bakalářské práce
- adresář *blueFinger*  
Obsahuje výslednou aplikaci určenou pro koncového uživatele.
- adresář *source\blueFingerServer*  
Obsahuje zdrojové kódy serverové části (Java), včetně projektových souborů k vývojovému prostředí NetBeans.
- adresář *source\blueFingerClient*  
Obsahuje zdrojové kódy klientské části (J2ME), včetně projektových souborů k vývojovému prostředí NetBeans.
- adresář *source\paper*  
Obsahuje zdrojové kódy k technické zprávě (L<sup>A</sup>T<sub>E</sub>X).
- adresář *doc\client*  
obsahuje dokumentaci vygenerovanou nástrojem Javadoc, titulní strana je v souboru *index.html*.
- adresář *doc\server*  
obsahuje dokumentaci vygenerovanou nástrojem Javadoc, titulní strana je v souboru *index.html*.

## Příloha B

# Uživatelská příručka

Tento manuál ukazuje, jak aplikaci správně spustit a jak s ní pracovat. Na přiloženém CD v adresáři *blueFinger* najdeme výslednou aplikaci určenou koncovému uživateli.

### B.1 Než začneme

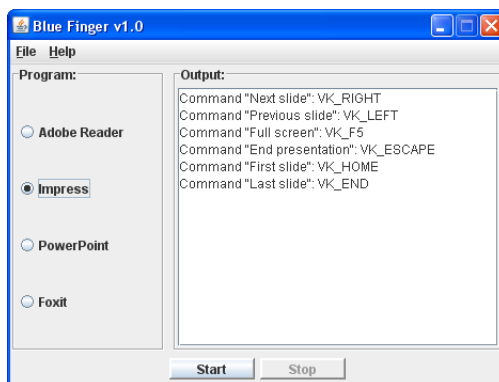
Pokud chceme aplikaci spustit, musíme se nejdříve ujistit, že počítač, na kterém poběží serverová část, je vybaven operačním systémem Windows XP a zařízením Bluetooth. Dále zde musí být nainstalovaná Java, pokud není, v adresáři najdeme její instalátor pro 32 i 64-bitovou verzi operačního systému.

Mobilní telefon musí obsahovat podporu mobilní Javy a Bluetooth. Tyto zařízení musí být spárována, tedy v každém zařízení musí být to druhé přidáno mezi důvěryhodné.

Poté již nic nebrání odeslání \*.jar a \*.jad souboru z adresáře *blueFinger\mobile* do mobilního telefonu a nainstalování aplikace.

### B.2 Spuštění serveru

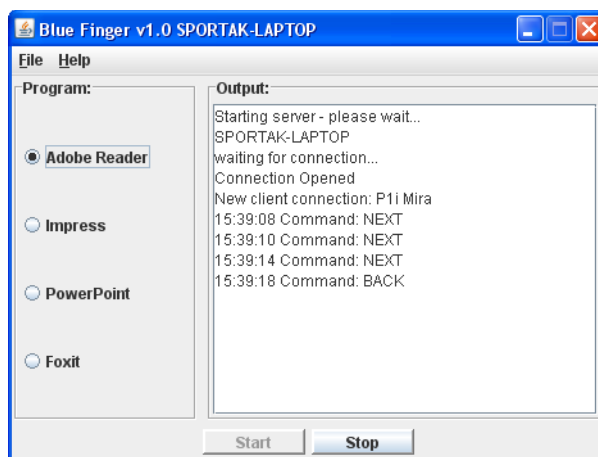
Serverová část se spouští souborem *blueFinger.bat*. Poté se zobrazí okno aplikace společně s oknem příkazového řádku. Toto okno není s aplikací nijak spjato, a proto může být zavřeno. Okno po spuštění tedy vypadá následovně.



Obrázek B.1: Okno serveru s výpisy ovládacích prvků.

V horní části obrázku B.1 můžeme vidět jednoduché menu, v kterém je možnost vybrat si, zda po změně prezentačního programu chceme vidět jeho ovládání. V levé části vidíme seznam podporovaných prezentačních programů, z kterých si uživatel vybírá. V pravé části vidíme okno s kontrolními výpisy programu. Pod nimi se nachází tlačítka pro spuštění a zastavení běhu serveru.

Po stisku tlačítka Start uživatel spustí běh serveru, který bude vyčkávat na připojení klienta. Po jeho připojení začne jeho příkazy pro kontrolu vypisovat do okna, viz obrázek B.2.



Obrázek B.2: Okno běhu serveru

Pro zastavení běhu serveru uživatel stiskne tlačítka Stop a tím se ukončí komunikace s klientem. Uživatel ovšem klienta musí ukončit sám.

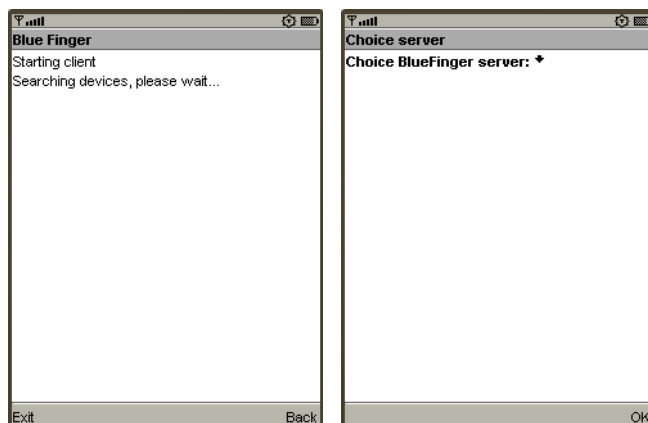
### B.3 Spuštění klienta

Pro spuštění klienta je nutné, aby server již běžel. Po spuštění klientské aplikace se již začnou vyhledávat všechna dostupná bluetooth zařízení v okolí jak můžeme vidět na obrázku B.3. Protože na každém mobilním telefonu může aplikace vypadat jinak, příklady obrazovky uvádím z emulátoru Javy. Tato obrazovka slouží zároveň i jako obrazovka pro kontrolní výpisy.

Jakmile vyhledávání Bluetooth zařízení doběhne, je uživateli nabídnuto, aby jedno z nich zvolil a na něm se bude vyhledávat služba serveru. Pokud uživatel zvolí zařízení špatně, komunikace se nenaváže. Okno s nabídkou zařízení v okolí by mohlo vypadat následovně. Na obrázku B.3, ovšem žádné zařízení v okolí dostupné není, pokud ano, bylo by v nabídce zahrnuto.

Po výběru serveru se obrazovka zařízení přepne do následujícího zobrazení B.4, kde můžeme vidět, které klávesy znamenají přechod mezi snímky. Zároveň tato oblast působí i jako tlačítka dotykového displeje, je-li jím mobilní telefon vybaven.

Jak můžeme vidět v pravé části obrázku B.4, aplikace obsahuje jednoduché menu, jejíž položky nabízí ovládací příkazy pro zobrazení prezentace přes celou obrazovku(1) a její



Obrázek B.3: Okno vyhledávání serverů a okno výběru serveru



Obrázek B.4: Okno běhu aplikace a menu

zrušení (2), položku pro přechod na první snímek prezentace(3) a poslední snímek prezentace (4), další položka (5) dovoluje prohodit ovládací prvky přechodu mezi následujícím a předchozím snímkem. Nabídka menu (6) úplně vypne nebo zapne zvuk na počítači, kde běží server. Poslední položka přechází z ovládací režimu na obrazovku s kontrolními výpisy, odkud je možné vrátit se zpátky.

Přechod mezi dalším a předchozím snímkem, je na běžném mobilním telefonu viz. [B.5](#) řešen pomocí stisku kláves 1,3,7 resp. 3,6,9. Je na uživateli, jakou řadu tlačítek si zvolí pro přechod na další snímek. U mobilních telefonů vybavených dotykovým displejem je možné ovládat přechody pomocí něj. Při stisku klávesy (nebo displeje) se část obrazovky pro vybraný pokyn zvýrazní, jak je vidět na obrázku [B.5](#). Pro zvýšení/snížení hlasitosti počítače slouží klávesy 2/8.

## B.4 XML soubor

Uživatel si sám může nadefinovat ovládání prezentačních programů pomocí XML souboru. Jeho editace vyžaduje pouze textový editor a trochu zručnosti. Pro editaci otevřete sou-



Obrázek B.5: Schema pro ovládání přechodu mezi snímky

bor ve složce aplikace *xml\controls.xml*. Vyberte jednu část od `<program>` do `</program>` a pouze zbývá upravit část tagu `<key code="zkratka">` tento tag se po sobě může opakovat libovolněkrát, záleží pouze na počtu kláves pro určitý příkaz. Pořadí příkazů je pevně dané, stejně jako jeho počet. Uživatel tedy může upravit pouze klávesové zkratky, které najde zde [8]. Úprava tohoto souboru je ale určená pouze pro zkušené uživatele.

Jeho špatnou editací by mohly nastat problémy, proto před editací soubor zálohujte. V příloze C lze nalézt příklad části XML souboru pro jeden prezentační program.

## Příloha C

# Příklad XML souboru

```
<root>
<program name="Foxit" id = "0">
  <action name="next" >
    <key code="VK\_RIGHT" ></key>
  </action>
  <action name="back">
    <key code="VK\_LEFT" ></key>
  </action>
  <action name="start">
    <key code="VK\_F11" ></key>
  </action>
  <action name="close">
    <key code="VK\_ESCAPE" \></key>
  </action>
  <action name="Home">
    <key code="VK\_HOME"></key>
  </action>
  <action name="END">
    <key code="VK\_END"></key>
  </action>
</program>
</root>
```