

# Využití J2ME při vývoji telemedicínských aplikací pro klasické mobilní telefony

Josef Špidlen

*EuroMISE centrum – Kardio, Ústav informatiky AV ČR  
Pod Vodárenskou věží 2, 182 00 Praha 8*

spidlen@euromise.cz

**Souhrn.** V posledních letech zažíváme značný pokrok v oblasti telekomunikačních služeb i informačních technologií. Mobilní telefon se stává nepostradatelným pomocníkem poskytující stále širší paletu služeb a implementující stále častěji běhovou podporu pro aplikace v jazyce Java. V rámci výzkumu v EuroMISE centru jsme se tedy přirozeně zabývali otázkou, zda a jak by bylo možné využít mobilní Javu pro vývoj telemedicínských aplikací. Článek přináší přehled nejmenší edice jazyka Java – Java 2 Micro Edition (J2ME) – a shrnuje možnosti specifikací „Connected Limited Device Configuration” (CLDC) a “Mobile Information Device Profile” (MIDP). Specifikace jsou analyzovány převážně ve vztahu k možnostem využitelným v telemedicínských aplikacích. V článku je podán stručný návod jak vyvinout telemedicínskou aplikaci ve formě Midletu, jak spolupracovat s uživatelským rozhraním mobilního telefonu a naznačeny jsou možnosti v komunikaci se serverovou aplikací na internetu. Představena je pilotní J2ME aplikace formalizující lékařská doporučení pro léčbu hypertenze – „1999 WHO/ISH Hypertension Guidelines“, kterou si čtenář může sám vyzkoušet na svém mobilním telefonu. Na závěr jsou diskutována omezení, se kterými se setkáváme u většiny současných mobilních telefonů, a nastíněny jsou možnosti přinášené novými verzemi J2ME specifikací, které jsou postupně implementovány v nejmodernějších mobilních zařízeních na trhu.

**Klíčová slova:** Java, J2ME, telemedicina, lékařská doporučení, hypertenze

**Abstract.** Špidlen J.: J2ME Usage in Telemedicine Applications Development for Classical Mobile Phones.

Lately we have been undergoing a significant progress in the field of telecommunication services and information technologies. A mobile phone becomes an indispensable tool permanently providing a wider range of services and more frequently implementing runtime support for applications in the Java language. In the research frame of the EuroMISE Centre we naturally dealt with the question whether and how it would be possible to utilize mobile Java for telemedicine applications development. The paper brings an overview of the smallest Java edition – Java 2 Micro Edition (J2ME) – and it also summarizes possibilities of the „Connected Limited Device Configuration” (CLDC) and “Mobile Information Device Profile” (MIDP) specifications. The specifications are analyzed mostly in relation to the possibilities applicable in telemedicine applications. Brief instructions how to develop a telemedicine application in the Midlet form and how to work with a mobile phone user interface are described. Possibilities of communication with the server application on Internet are outlined. The pilot J2ME application formalizing medical guidelines for hypertension treatments – „1999 WHO/ISH Hypertension Guidelines“ is introduced. Readers may try this application by themselves. Restrictions met in most contemporary mobile phones are discussed in conclusions. There are also described possibilities brought by new J2ME specification versions, which are gradually implemented in the latest mobile devices on the market.

**Keywords:** Java, J2ME, Telemedicine, Medical Guidelines, Hypertension

## 1. Úvod

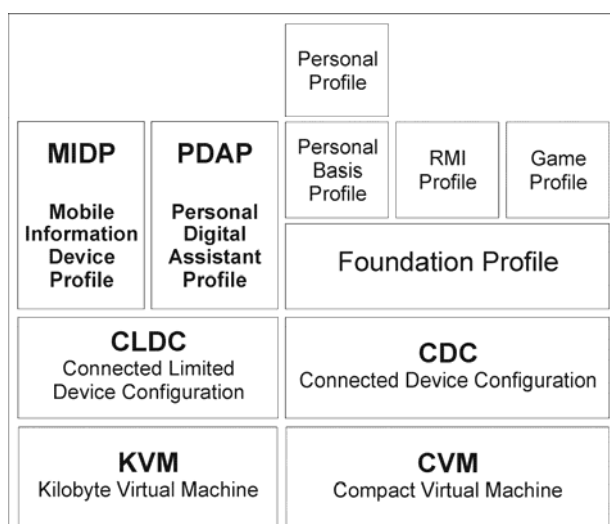
V dnešní době jen zřídka potkáte člověka, který by nevlastnil mobilní telefon. Stále více lidí si bez telefonu svůj život neumí ani představit. Navíc každý vidí, jak velký a rychlý pokrok v oblasti mobilní telekomunikace během posledních 5 až 10 let zažíváme. Vývoj v oblasti telekomunikačních služeb jde ruku v ruce s vývojem v oblasti informačních technologií. Výpočetní výkon mikroprocesorů používaných běžně v mobilních telefonech mnohonásobně

předčí výkon počítačů řídicích první let do vesmíru. Dnešní mobilní telefony již nejsou jen prostředky k telefonování. Často integrují diář, kalkulačku, budík, stopky, záznamník, fotoaparát a mnoho her a dalších aplikací, umožňují přístup k internetu pomocí WAPu, HTML prohlížeče nebo emailového klienta, podporují přenosy pomocí GPRS či HSCSD a nabízejí přímé propojení s počítačem nejen kabelem, ale i bezdrátově pomocí IrDA nebo Bluetooth. Vystává tedy přirozeně otázka, zda by dnes již běžných mobilních telefonů nebylo možné využít jako platformy pro telemedicínské aplikace.

V rámci telemedicínských projektů v Evropském centru pro medicínskou informatiku, statistiku a epidemiologii (EuroMISE centru) jsme se touto otázkou také zabývali. Původní výzkum [1, 2] byl orientován převážně na možnosti přístupu k elektronickému zdravotnímu záznamu z mobilních zařízení. Pro přístup bylo využíváno jednak HTML či WAP prohlížečů v mobilních telefonech, jednak podpory .NET Compact Frameworku na platformě Pocket PC 2002, například na T-Mobile MDA zařízeních. Tato zařízení ovšem nejsou zdaleka tak hojně rozšířená jako „obyčejné“ mobilní telefony. Na druhou stranu, většina mobilních telefonů v dnešní době uvádí podporu jazyka Java, a tedy v tomto článku se soustředíme na možnosti využití této mobilní Javy pro vývoj telemedicínských aplikací.

## 2. Java v mobilních telefonech

Slovo Java je dnes skloňováno ve všech pádech v souvislosti s mobilními telefony. Pojem Java často ztotožňuje objektově-orientovaný programovací jazyk Java s knihovnamí standardních tříd a sadami standardních aplikačních rozhraní. Tyto sady knihoven a rozhraní existují ve 3 základních edicích, Java2 Enterprise Edition (J2EE), Java2 Standard Edition (J2SE) a Java2 Micro Edition (J2ME). Jak by asi každý dle názvu hádal, z hlediska mobilních telefonů je pro nás relevantní právě J2ME, která vznikla s cílem sjednotit podporu Javy v malých zařízeních, zahrnujících nejen mobilní telefony, není J2ME jedna komplexní specifikace, ale člení se na různé konfigurace. Konfigurace určuje základní sadu knihoven a vlastností přístroje a je dále upřesňována tzv. profilem. Základní členění J2ME je zobrazeno na obrázku číslo 1. Text jednotlivých specifikací je k dispozici ke stažení na [www](http://www) stránkách [3].



Obr. 1 Základní dělení J2ME.

## 2.1. CLDC

Pro účely telemedicínských aplikací na mobilních telefonech je pro nás relevantní levá část obrázku. Konfigurace CLDC je zaměřena právě na malá zařízení, která musí obsahovat alespoň 128 kB stálé paměti, jejíž obsah musí zůstat zachován i po vypnutí zařízení (ale nemusí být aplikacím přístupný k zápisu), a 32 kB nestálé paměti, která musí být k dispozici virtuálnímu stroji při běhu. Zařízení musí disponovat minimálně 16-ti bitovým procesorem (RISC/CISC) o taktovací frekvenci alespoň 16 MHz. Z pohledu Java knihoven specifikuje CLDC jen malou část ze standardních balíčků `java.lang`, `java.io` a `java.util`. Navíc přidává nový balíček `java.microedition.io` pro vstupně/výstupní operace u tříd, kde je odlišnost od standardní edice příliš velká. Navíc specifikace CLDC explicitně vynechává (zakazuje) některé prvky ze standardní Javy. Co programátora v první řadě překvapí je nemožnost použití výpočtů s plovoucí desetinnou čárkou. Dále je třeba počítat se zjednodušenou destrukcí objektů, protože neexistuje `Object.finalize()`, a rovněž podpora zpracování chyb je omezena kvůli absenci třídy `java.lang.Error` a jejích potomků.

## 2.2. MIDP

Konfiguraci CLDC upřesňují 2 profily: PDAP, určený převážně pro zařízení typu PDA, a MIDP, specifikující použití na nejmenších zařízeních jakými jsou právě mobilní telefony. K hardwarové specifikaci přidává MIDP požadavek na minimální velikost displeje 96 x 54 pixelů a na možnost ovládat zařízení buď klávesami nebo dotykem obrazovky. Pro potřeby ukládání dat aplikací je zde vyžadováno aspoň 8 kB stálé paměti. Typický telefon poskytuje aplikaci za běhu několik desítek až stovek kB paměti, což není mnoho a je tedy nutné psát kód s ohledem na úsporu paměti. K balíčkům obsaženým v CLDC přidává MIDP další:

- `javax.microedition.rms` – prostředky pro správu trvalých dat
- `javax.microedition.midlet` – obdoba klasické třídy „appletu“
- `javax.microedition.io` – k CLDC přidána podpora HTTP spojení
- `javax.microedition.lcdui` – práce s uživatelským rozhraním

Skoro všechny mobilní telefony, kromě některých komunikátorů firmy Nokia, spadají do kategorie MIDP. Bohužel, podpora a vlastnosti jednotlivých telefonů se liší a tedy není možné spolehnout se na to, že MIDP aplikace určená pro jeden telefon bude bez problémů fungovat i na telefonu jiném. Přestože specifikace vznikly právě za účelem sjednocení funkčnosti a rozhraní jednotlivých telefonů, praktické zkušenosti vývojářů ukazují, že i když výrobci telefonů implementují „tutéž“ J2ME specifikaci, výsledná implementace se na jednotlivých telefonech liší. Většina telefonů v současné době implementuje CLDC verze 1.0 a MIDP verze 1.0. V těchto specifikacích bylo pravděpodobně pozapomenuto na ovládání zvuku, přístup k SMS, MMS a některé další užitečné funkce, což výrobci telefonů řeší přidáním vlastních rozšiřujících tříd, které tuto funkcionalitu poskytují. Jakmile však vývojář použije rozšiřující třídy, stává se aplikace nepřenositelnou. Pro potřeby telemedicínských aplikací je závažným problémem nedostatečná podpora přenosových služeb. Třída `URLConnection` postačí na jednoduché prohlížení HTML a WAP stránek, ale pro plnohodnotný přístup k internetu to není mnoho. Způsob, jakým je možno přistupovat k internetu pomocí HTTP protokolu, bude nastíněn dále. Mnoho nedostatků ve specifikaci MIDP 1.0 řeší specifikace MIDP verze 2.0, kterou v současné době implementuje jen několik zařízení. Problémem je, že specifikace MIDP 2.0 je poměrně obsáhlá a její implementace vyžaduje nadprůměrné hardwarové požadavky. Na trhu v naší republice existují pouze dvě zařízení podporující MIDP verze 2.0 a jsou to poměrně drahé mobilní telefony Sony Ericsson P900 a Nokia 6600.

### 2.3. Další rozšíření

Některá zařízení se snaží poskytnout širší funkčnost, než specifikuje CLDC a MIDP verze 1.0, ale implementovat přímo MIDP verze 2.0 není jednoduchou záležitostí. Různí výrobci tedy přidávají rozličné sady tříd spolupracující s jejich zařízeními. Ve snaze unifikovat tyto balíčky vznikají v rámci specifikace J2ME tzv. volitelné balíčky (Optional Packages). Těchto dílčích specifikací je v současné době celkem 14, z nichž nejrozšířenější jsou Mobile Media API (MMAPI), Wireless Messaging API (WMA) a Java APIs for Bluetooth (BTAPI). Kompletní seznam zařízení podporujících J2ME se pokouší udržovat firma Sun [4]. Tento seznam se však vyvíjí příliš rychle a proto nemůžeme mít firmě Sun za zlé, že není zcela aktuální a kompletní. Navíc, prakticky všichni výrobci uvádějí pouze informaci o tom, že telefon podporuje „Javu“, ale zjistit jaké standardy jsou konkrétně podporovány je velkým oříškem. Přesto se podařilo v tabulce číslo 1 uvést alespoň základní přehled mobilních telefonů dostupných v současné době na českém trhu a podporujících specifikace MIDP a CLDC alespoň ve verzích 1.0.

**Tab. 1:** Mobilní telefony s podporou J2ME běžně v prodeji v České Republice

Výrobce	Model	Software	Displej (pix.)
Motorola	T720	MIDP 1.0, CLDC 1.0	120x160 / 12 bitů
Motorola	V60i	MIDP 1.0, CLDC 1.0	96 x 64 / 1 bit
Nokia	3100	MIDP 1.0, CLDC 1.0, WMA 1.0	128x128 / 12 bitů
Nokia	3200	MIDP 1.0, CLDC 1.0, WMA 1.0	128x128 / 12 bitů
Nokia	3300	MIDP 1.0, CLDC 1.0, WMA 1.0, MMAPI 1.0	128x128 / 12 bitů
Nokia	3410	MIDP 1.0, CLDC 1.0	96x65 / 1 bit
Nokia	3510i	MIDP 1.0, CLDC 1.0	96x65 / 12 bitů
Nokia	3650	MIDP 1.0, CLDC 1.0, WMA 1.0, MMAPI 1.0	176x208 / 12 bitů
Nokia	5100	MIDP 1.0, CLDC 1.0	128x128 / 12 bitů
Nokia	6100	MIDP 1.0, CLDC 1.0	128x128 / 12 bitů
Nokia	6220	MIDP 1.0, CLDC 1.0, WMA 1.0	128x128 / 12 bitů
Nokia	6310i	MIDP 1.0, CLDC 1.0	96x65 / 1 bit
Nokia	6600	MIDP 2.0, CLDC 1.0, WMA 1.0, MMAPI 1.0, BTAPI 1.0	176x208 / 16 bitů
Nokia	8910i	MIDP 1.0, CLDC 1.0	96x65 / 12 bitů
Nokia	9210i	MIDP 1.0, CLDC 1.0, JavaPhone 1.0, PersonalJava 1.1.1	640x200 / 12 bitů
Siemens	C55	MIDP 1.0, CLDC 1.0, MMAPI 1.0	101x64 / 1 bit
Siemens	S55	MIDP 1.0, CLDC 1.0, MMAPI 1.0	101x80 / 8 bitů
Sony Ericsson	P800	MIDP 1.0, CLDC 1.0, PersonalJava 1.1.1	208x320 / 12 bitů
Sony Ericsson	P900	MIDP 2.0, CLDC 1.0, PersonalJava 1.1.1, WMA 1.0, BTAPI 1.0	208x320 / 16 bitů
Sony Ericsson	T610	MIDP 1.0, CLDC 1.0, MMAPI 1.0	128x160 / 16 bitů
Sony Ericsson	T630	MIDP 1.0, CLDC 1.0, MMAPI 1.0	128x160 / 16 bitů
Sony Ericsson	Z600	MIDP 1.0, CLDC 1.0, MMAPI 1.0	128x160 / 16 bitů

Bohužel je třeba říci, že spoléhat se na přenositelnost J2ME aplikací z jednoho telefonu na druhý bez patřičného otestování není možné ani v případě, že telefony podporují stejné konfigurace, profily i rozšiřující balíčky. Vývojář se musí smířit s tím, že telefon ohlásí výjimku z důvodu nenalezení některé metody ve třídě, kde by tato metoda být měla, nebo že třeba vůbec nenalezne třídu, protože ji výrobce místo standardního javax balíčku umístil proprietárně zcela jinam. Chceme-li, aby naše aplikace fungovala univerzálně na co nejvíce mobilních telefonech, měli bychom se vystříhat použití jakýchkoli nestandardních nebo málo rozšířených balíčků a pokud možno se omezit na základy specifikované v MIDP a CLDC verze 1.0.

### 3. Vývoj telemedicínské aplikace

Pro vývoj J2ME telemedicínské aplikace je třeba začít vývojovým prostředím. Vhodné je instalovat J2SDK a J2ME Wireless Toolkit, který lze integrovat do některých standardních vývojových prostředí pro jazyk Java (např. Forte či Borland JBuilder). Jinou možností je využít například Borland Mobile Studio, které vše potřebné integruje samo o sobě. Pro testování je dále vhodné instalovat emulátory mobilních telefonů, které lze zpravidla získat od výrobců.

#### 3.1. Midlet

Tak jako je základem Java aplikací vložených do HTML stránek tzv. Applet, je základem Java aplikací na mobilních telefonech tzv. Midlet, který musí rozšiřovat abstraktní třídu `javax.microedition.midlet.MIDlet`. Midlet běží z bezpečnostních důvodů v samostatném paměťovém prostoru a může se nacházet ve 3 různých stavech: aktivní, pasivní nebo zrušený. Přejít mezi stavy řídí aplikační manažer telefonu. Není možné, aby bylo více J2ME aplikací aktivních současně, tj. pouze jeden Midlet může být aktivním.

Pouze aktivní Midlet má přístup na displej a dokáže detekovat uživatelské akce. Princip detekce stlačení funkční klávesy je podobný jako detekce událostí u standardní edice Javy. Je třeba implementovat rozhraní `CommandListener` sestávající z jediné metody `commandAction(Command c, Displayable d)`. V inicializaci příslušného objektu pak pomocí volání `addCommand(Command c)` přidáme postupně všechny příkazy, na které budeme chtít nějak reagovat. Každý příkaz obsahuje krátký textový popis, číslo a identifikaci funkční klávesy, kterou má být vyvolán. Funkční klávesy jsou pojmenované `BACK`, `CANCEL`, `OK` apod. Umístění těchto funkčních kláves se na jednotlivých mobilních telefonech liší. Navíc, přidáme-li více než dva příkazy, telefon sám jeden z nich vybere a přiřadí mu pravou nebo levou funkční klávesu. Druhé funkční klávese ve skutečnosti přiřadí malé menu, ve kterém bude uživateli umožněno vybrat ze všech ostatních příkazů. Aby byly příkazy správně zpracovány, musíme také objektu starajícimu se o výstup na displej dát vědět, který objekt bude sloužit pro zpracování příkazů. K tomuto účelu typicky využijeme metodu `setCommandListener()`.

#### 3.2. Displej

Základní třídou, která spravuje displej telefonu, je třída `Display`. V rámci jednoho Midletu existuje právě v jedné instanci, kterou lze získat použitím statické metody `getDisplay()`. Objekty, které se zobrazují na displeji, musí implementovat rozhraní `Displayable`. Principiálně existují dvě možnosti, jak kreslit a psát na displej telefonu. Nízkoúrovňovou možností je dědit od abstraktní třídy `Canvas` a pak přímo přistupovat na displej. Tento přístup je vhodný při tvorbě různých her, které v současné době tvoří přes 90% J2ME aplikací na mobilních telefonech. Kromě značného množství vlastního kódu nutného pro tento přístup je zde ještě jedna velká nevýhoda. Různé telefony se chovají různě, často ignorují příkazy k zarovnání na různé strany (`BOTTOM`, `RIGHT`, `CENTER`, ...) a neřídka odmítají sdělit, jakou velikost (`getWidth`, ...) má displej. Když na jednom telefonu aplikace funguje, klidně může na jiném „psát přes sebe“. Je tedy velmi obtížné vytvořit přenositelnou aplikaci.

Při tvorbě telemedicínských aplikací si ale naštěstí zpravidla vystačíme s vysokoúrovňovým přístupem. Využijeme jednoho z potomků abstraktní třídy `Screen`, tedy třídy `Alert`, `Form`, `List` nebo `TextBox`. Právě třída `Form` představující obecný formulář bude zpravidla naším

základem. Do tohoto formuláře můžeme vkládat jednoduché ovládací prvky podobným způsobem (metodou `append`), jak jsme zvyklí při tvorbě aplikací určených pro osobní počítače PC. Rozdíl je v tom, že nspecifikujeme žádný „layout“, telefon sám určuje umístění každého následujícího prvku dle svých možností. Typicky prvky klade jednoduše jeden pod druhý. Základní ovládací prvky, které zde můžeme použít, jsou uvedeny v tabulce číslo 2.

**Tab. 2:** Ovládací prvky, které lze vložit na formulář mobilního telefonu.

<b>Třída</b>	<b>Popis</b>
<code>DataField</code>	Vstupní pole speciálně uzpůsobené pro vstup údaje typu datum nebo čas.
<code>Choice</code>	Dvoustavový přepínač.
<code>ChoiceGroup</code>	Skupina přepínačů, z nichž pouze jeden nabývá pozitivní hodnoty.
<code>Image</code>	Obrázek, typicky ve formátu jpeg, bmp, png nebo gif. Podporované typy (a podtypy) závisí na druhu telefonu.
<code>Label</code>	Textový popisek.
<code>List</code>	Vložený seznam dalších položek.
<code>TextField</code>	Textové pole. Lze specifikovat, zda vstupem má být textový řetězec, celé číslo, telefonní číslo, url adresa apod. Telefon se této volbě přizpůsobí a kontroluje vstupní formát.
<code>Ticker</code>	Pruh či posouvátko naznačující průběh zpracování operace případně sloužící k interaktivnímu zadání hodnoty nastavením polohy posouvátka.

Nastavovat lze různé vlastnosti těchto ovládacích prvků, ale je nutné počítat s tím, že některé telefony nebudou vše zcela respektovat. Typickým příkladem je změna fontu, kterou některé telefony provedou, ale jiné ignorují.

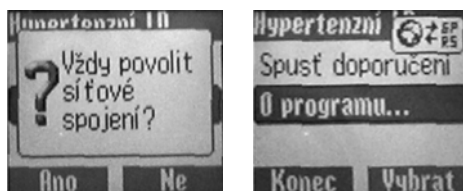
### 3.3. Spojení se serverovou aplikací

I když lokální J2ME aplikace výrazně rozšiřují možnosti mobilního telefonu, ještě zajímavější mohou být aplikace s připojením k serveru. Představit si můžeme mobilní klientské aplikace elektronického zdravotního záznamu či například aplikace poskytující telekonzultaci s odborníky různých specializací. Jak již bylo nastíněno, profil MIDP 1.0 nám zajišťuje pouze HTTP spojení. Důvodem proč pouze HTTP spojení, uváděným autory specifikace, je skutečnost, že některé mobilní sítě jsou implementovány způsobem, který by se jen těžko přizpůsoboval připojení pomocí TCP/IP socketů. V případě HTTP spojení není tolik významné, jaké jsou použity nižší vrstvy sítě. Navíc bohužel nemáme zaručeno, kolik spojení může být otevřeno zároveň. Například většina telefonů značky Nokia umožňuje pouze jedno otevřené spojení a je tedy vhodné nespoléhat se na to, že jich budeme mít k dispozici více. Jak otevřít HTTP spojení a načíst soubor z internetu ukazuje následujících několik řádek v jazyce Java.

```
public static String connect(String url) throws IOException {
    HttpURLConnection cn = null;
    InputStream is = null;
    int ch;
    try {
        cn = (HttpURLConnection) Connector.open(url);
        cn.setRequestMethod(HttpURLConnection.GET);
        is = cn.openInputStream();
        StringBuffer buf = new StringBuffer();
        while ((ch = is.read()) != -1) buf.append((char)ch);
        return buf.toString();
    } finally {
        if (is != null) is.close();
        if (cn != null) cn.close();
    }
}
```

}  
}

Protokol HTTP nám kromě načítání dat ze serveru umožňuje i jejich odesílání. Využit můžeme například zakódování proměnných do url adresy či klasickou HTTP POST metodu. Důležité je upozornit, že ladit aplikace vytvářející HTTP spojení není možné pomocí emulátorů. I když máte své PC připojeno k internetu, emulátory tuto komunikaci nepodporují a je tedy nutné ladit aplikaci pouze s použitím opravdového telefonu. Aby nebylo možné vytvořit aplikaci, která bude komunikovat bez uživatelského vědomí, zeptá se telefon uživatele explicitně, zda vytvoření spojení povolit. Vzhledem k stále obecně nemalým cenám za mobilní datové služby zobrazuje telefon po celou dobu připojení symbol, který uživatele na komunikaci upozorňuje. Jakým způsobem toto provádí například Siemens S55 je znázorněno na obrázku číslo 2.



Obr. 2: Mobilní telefon Siemens S55 signalizující spojení.

## 4. Hypertenzní lékařská doporučení jako J2ME aplikace

Lékařská doporučení jsou systematicky vytvářené návody, které usnadňují lékařům rozhodování v rozmanitých medicínských situacích. Tato doporučení jsou lékařům obvykle předkládána ve formě vytištěného textu. S rozvojem informačních a komunikačních technologií se začíná projevovat potřeba vytvářet lékařská doporučení také v elektronické podobě, která by umožňovala informaci obsaženou v lékařských doporučeních interaktivně využívat. V EuroMISE centru jsme se již zabývali otázkami, jak reprezentovat doporučení elektronicky tak, aby bylo možné aplikovat algoritmy pro odvozování závěrů ze znalostí obsažených v lékařských doporučeních a ze známých dat o pacientovi. Jako pilotní J2ME aplikaci jsme implementovali formalizaci lékařských doporučení „1999 WHO/ISH Guidelines for the Management of Hypertension“ [5]. Je nutné upozornit, že lékařská doporučení jsou určena především lékařům a slouží pouze doplňkově k podpoře jejich rozhodování. Zodpovědnost za každé rozhodnutí leží vždy na lékaři. Pilotní J2ME aplikace hypertenzních lékařských doporučení je určena především k demonstraci možností Java aplikací na mobilních telefonech.

### 4.1. Algoritmus hypertenzních doporučení

Algoritmus pilotní J2ME aplikace implementuje postup doporučený pro léčbu vysokého tlaku uvedený v [5]. Kromě zjištění aktuálního krevního tlaku je nutné zadat hned v první fázi další údaje jako jsou věk, cholesterol, pohlaví apod. Příklad vstupu zobrazuje obrázek číslo 3 demonstrující aplikaci na telefonu Siemens S55. Ze zadaných informací program nejprve zjistí stupeň hypertenze a analyzuje rizikové faktory kardiovaskulárního onemocnění. Dále, s přihlédnutím na tzv. poškození cílových orgánů a přidružené klinické stavy, spočítá stupeň rizika kardiovaskulárního onemocnění v následujících deseti letech. V případě zvýšeného krevního tlaku je pak na základě ostatních údajů doporučeno sledování pacienta po určitou dobu nebo nasazení medikamentózní léčby. Pokračujeme-li léčbou, vyžádá si telefon zadání údajů pro stanovení indikací a kontraindikací různých skupin léčiv. Dle absolutních i relativních indikací a kontraindikací jsou léčiva seřazena od nejvhodnějších po ta nejméně

vhodná. Zdají-li se dvě skupiny léčiv v tomto směru ekvivalentní, preferuje algoritmus levnější léky. V tomto okamžiku ještě jednou upozorňujeme, že aplikace slouží jen jako doplňková podpora rozhodování lékařů a jen lékař může komplexně posoudit stav pacienta.



Obr. 3: J2ME aplikace lékařských doporučení pro léčbu hypertenze na telefonu Siemens S55.

#### 4.2. Implementace aplikace hypertenzních doporučení

Ve stejném duchu, jako byla architektura J2ME aplikací popsána obecně, je navržena i pilotní J2ME aplikace s hlavním Midletem `HypertensionGD`. Po startu je aktivován potomek třídy `List`, který dá na výběr spuštění aplikace doporučení nebo zobrazení dialogu „O programu...“. První možnost aktivuje objekt třídy `HTGDForm`, druhá `AboutBox`. Obě tyto třídy jsou potomky třídy `Form`. Na formuláře jsou postupně vloženy ovládací prvky a přidány jsou reakce na funkční tlačítka telefonu. V případě postupu algoritmu lékařských doporučení je procházeno ještě několika formuláři sloužícími k zadání dalších potřebných dat o pacientovi. Po tuto dobu je doplňován objekt třídy `PatientDetails` představující aktuálního pacienta a informace o něm. Ve chvíli, kdy je potřeba doporučit medikamentózní léčbu, jsou použity objekty třídy `DrugClass`. Tyto objekty jsou vytvořeny pro různé druhy léčiv používaných k léčbě hypertenze, konkrétně pro diuretika, beta-blokátory, alfa-blokátory, blokátory kalciového kanálu, ACE inhibitory a antagonisty angiotensinu II. U těchto objektů jsou v závislosti na stavu pacienta zjištěny relativní i absolutní indikace a kontraindikace a objekty jsou seříděny dle vhodnosti známým algoritmem „Bubble Sort“. Celkový výsledek je zobrazen uživateli na výsledném formuláři.

#### 4.3. Jak nahrát J2ME aplikaci do telefonu

Aplikace se skládá ze dvou základních souborů, které mají strukturu danou specifikací profilu MIDP. Jsou to deskriptor aplikace – Java Application Descriptor (JAD soubor) a soubor s aplikací samotnou ve tvaru JAR soubor. JAR soubor představuje ve skutečnosti archiv



formátu ZIP s pevně danou strukturou. Kromě Java tříd a ostatních zdrojů (například obrázků) obsahuje JAR soubor tzv. manifest, který musí být v adresáři META-INF a musí se jmenovat manifest.mf. Tento textový soubor musí obsahovat na každém řádku vždy dvojici – jméno atributu a jeho hodnotu oddělenou dvojtečkou. Atributů je definováno mnoho, ty povinné ukazuje následující příklad:

```
MIDlet-Name: Hypertension Guidelines  
MIDlet-Icon: /icons/logo.png  
MIDlet-Vendor: EuroMISE Centre - Cardio  
MIDlet-1: Hypertension GD, /icons/logo.png, cz.spidlen.phone.gtgd.HtGD  
MIDlet-Version: 1.0  
MicroEdition-Configuration: CLDC-1.0  
MicroEdition-Profile: MIDP-1.0
```

Stejnou strukturu jako manifest má i deskriptor aplikace, ve kterém jsou podobné i povinné atributy. Rozdíl je v tom, že v deskriptoru se neuvádějí atributy MicroEdition-Configuration a MicroEdition-Profile, ale navíc dva atributy přibývají. Jsou jimi MIDlet-Jar-Size a MIDlet-Jar-URL, které specifikují velikost JAR souboru v bytech a jeho url adresu.

Máme-li stručně popsány 2 základní soubory aplikace, není složité ji do telefonu nainstalovat. Existuje několik různých způsobů jak na to. Jestliže telefon podporuje připojení k počítači sériovým či USB kabelem, pomocí infračerveného portu nebo standardu Bluetooth, pak výrobce pravděpodobně dodává s telefonem i komunikační software, s jehož pomocí můžeme soubory jednoduše do telefonu nakopírovat. Dle konkrétního telefonu soubory buď uložíme do paměti pod složku pojmenovanou zpravidla „Java“ nebo jen odešleme na telefon, ten si je sám uloží a aplikace je tímto nainstalována.

Jestliže nemůžeme použít žádný z výše uvedených způsobů, zbývá univerzální možnost a tou je instalace s využitím wapu. Soubory JAR i JAD umístíme na HTTP server. Na tomto serveru musí být správně nastavený MIME-Type pro JAR i JAD soubory, aby je software v telefonu dokázal rozpoznat. Toto nastavení uvádí tabulka číslo 3.

**Tab. 3:** Nastavení MIME-Type pro instalaci J2ME aplikací z HTTP serveru.

Přípona souboru	MIME-Type
jar	application/java-archive
jad	text/vnd.sun.j2me.app-descriptor

Po umístění obou souborů na HTTP server aplikaci nainstalujeme pomocí wap prohlížeče mobilního telefonu. Některé telefony stahují JAR soubor přímo přes wap, jiné si otevírají přímé spojení k JAR souboru podle informace z MIDlet-Jar-URL v deskriptoru aplikace. Univerzálním řešením, které by mělo fungovat na všech telefonech, je vyplnit v tomto atributu úplnou adresu JAR souboru. Toto řešení jsme tedy také využili pro zpřístupnění pilotní J2ME aplikace hypertenzních lékařských doporučení široké veřejnosti. Na internetové adrese <http://www.euromise.cz/mobile> lze získat jak aplikaci samotnou, tak její zdrojové kódy v jazyce Java sloužící jako doplňující příklad k tomuto článku.

## 5. Závěr

V článku jsme se seznámili se základními možnostmi využití J2ME při vývoji telemedicínských aplikací pro mobilní telefony. Jestliže máte zájem se s touto technologií seznámit více do hloubky, můžeme doporučit jednu z knih [6] či [7]. Mobilní Java je zatím sice z více než 90% využívána pro vývoj rozličných her, ale z provedených analýz vyplývá, že i

v oblasti telemedicíny nalezne tato nejmenší edice jazyka Java své uplatnění. Na druhou stranu si musíme přiznat, že malý displej a jen několik vstupních kláves jsou a vždy budou omezujícími faktory pro komfortní a uživatelsky příjemné aplikace. Výrobci telefonů mají v tomto ohledu nelehkou situaci, uživatelé si žádají čím dál větší a barevnější displej a ne příliš malá tlačítka, ale celkově musí být telefon stále menší, lehčí a výkonnější.

Omezujícími faktory pro vývojáře telemedicínských J2ME aplikací jsou hardwarové i softwarové limity mobilních telefonů. Aplikace musejí být zatím hodně malé – přesná velikost a povolené množství spotřebované paměti i dalších zdrojů se u jednotlivých telefonů liší. Nicméně vzhledem k tomu, že se na našem trhu objevily již první 2 vlašťovky implementující profil MIDP verze 2.0, můžeme očekávat, že možnosti porostou velmi rychle. Se specifikací MIDP 2.0 přicházejí některá nízkoúrovňová rozhraní, například již zmiňované sockety či dokonce UDP datagramy. Nevýhoda stávajícího omezení na spojení pouze HTTP protokolem je například ta, že HTTP vylučuje inicializaci spojení ze strany serveru. Představíme-li si aplikaci, kdy server v nemocnici obdrží laboratorní výsledky pacienta a chce je sám předat dál aplikaci běžící na mobilním telefonu pacienta, nastává těžko řešitelný problém.

Vzhledem k rychlosti pokroku v oblasti telekomunikačních služeb i informačních technologií zajisté můžeme počítat s tím, že i MIDP verze 2.0 bude za nedlouho rozšířená v běžně používaných mobilních telefonech. Výrobci se navíc stále více snaží sjednotit podporu pro pokročilá uživatelská rozhraní (Personal Java), Bluetooth, IrDA, zprávy typu SMS, EMS, MMS i emailové a tak můžeme očekávat, že telemedicínské J2ME aplikace na mobilních telefonech se brzy stanou reálnou skutečností.

## **Poděkování**

Tato práce byla částečně podpořena projektem číslo LN00B107 MŠMT ČR.

## **Literatura**

1. Špidlen J., Štochl J., Semecký J., Hanzlíček P.: MUDR and Mobile Communication, Medical Informatics Europe, Proceedings CD, St. Malo 2003
2. Hanzlíček P., Špidlen J., Zvárová J.: The Internet in Connecting Electronic Health Record Mobile Clients, Technology and Health Care, Vol. 10, Num 6, 2002, ISSN: 0928-7329, pp. 502-503
3. Sun Microsystems, Inc.: Introduction to Mobility Java Technology, <http://wireless.java.sun.com/getstart>
4. Sun Microsystems, Inc.: J2ME Device List, <http://jal.sun.com/webapps/device/>
5. World Health Organization – International Society of Hypertension: Guidelines for the Management of Hypertension. Journal of Hypertension, 17, 1999, pp. 151-183.
6. Topley K.: J2ME in a Nutshell, ISBN: 0-596-00253-X, O'Reilly & Associates, Inc. 2002
7. Riggs R.: Programming Wireless Devices with the J2ME Platform, ISBN: 0201746271, Addison Wesley Publishing Company, 2001

## **Adresa pro korespondenci**

Josef Špidlen, EuroMISE centrum, ÚI AV ČR  
Pod Vodárenskou věží 2, 182 07 Praha 8  
email: [spidlen@euromise.cz](mailto:spidlen@euromise.cz)