

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE PRO PREZENTACI A MODIFIKACI DAT V PŘENOSNÝCH ZAŘÍZENÍCH

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PAVEL KUČERA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE PRO PREZENTACI A MODIFIKACI DAT V PŘENOSNÝCH ZAŘÍZENÍCH

APPLICATION FOR DATA PRESENTATION AND MODIFICATION IN MOBILE DEVICES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PAVEL KUČERA

VEDOUcí PRÁCE

SUPERVISOR

Mgr. ROMAN TRCHALÍK

BRNO 2008

Abstrakt

Tato práce se zabývá možnostmi využití přenosných zařízení v oblasti automatizace řízení. Zadání práce vychází z reálného požadavku společnosti působící v oblasti automatizace mít k dispozici nástroj pro vzdálené řízení technologického procesu prostřednictvím PDA či mobilního telefonu. Jsou analyzovány možnosti uživatelského softwaru v současných přenosných zařízeních s ohledem na předpokládané využití. Podle specifikace požadavků je proveden návrh a implementace systému typu klient–server založeného na technologii VNC. Serverem je linuxový PC server realizovaný pomocí Xinetd a Xvnc. Jako klienti slouží dva standardní VNC klienti implementovaní modifikací softwaru volně dostupného pod licencí GPL. Jeden je určen pro zařízení s operačním systémem Windows Mobile, druhý běží na zařízeních s podporou Javy.

Klíčová slova

mobilní telefon, mobil, PDA, smartphone, vzdálené řízení, přenosné zařízení, RFB, VNC, RFB protokol, řízení procesů, automatizace, měření a regulace, vizualizace, superserver, Xinetd, Xvnc, Xvnc server, server, klient, QuickControl

Abstract

This thesis is aimed to study the possibility of usage of mobile devices in the sphere of automation. One automation company needed to have in hand some PDA–based tool for a remote control of the technological process. There is analysis of ability of contemporary portable devices to run a application software. According to the specified requirements there is designed and implemented the client–server system based on the VNC technology. The server is a Linux–based PC server implemented using Xinetd daemon and the Xvnc. The two standard VNC clients were made. A free software available under the GPL license was modified to implement them. One runs on Windows Mobile operating system and the other one is a Java MIDlet.

Keywords

cell phone, PDA, smartphone, remote control, portable equipment, RFB, VNC, RFB protocol, process control, automatization, measurement and regulation, visualisation, superserver, Xinetd, Xvnc, Xvnc server, server, client, QuickControl

Citace

Pavel Kučera: Aplikace pro prezentaci a modifikaci dat v přenosných zařízeních, diplomová práce, Brno, FIT VUT v Brně, 2008

Aplikace pro prezentaci a modifikaci dat v přenosných zařízeních

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Mgr. Romana Trchalíka. Další informace mi poskytli Ing. Rostislav Fitz a Ing. Miroslav Fitz. Uvedl jsem všechny hlavní literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Kučera
19. května 2008

© Pavel Kučera, 2008.

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

Obsah

1	Úvod	3
2	Současný stav	4
2.1	Systémy měření a regulace	4
2.1.1	Schéma systémů měření a regulace	4
2.2	QuickControl	6
2.2.1	Vztah ke schématu systémů měření a regulace	7
2.2.2	Klíčové vlastnosti QuickControl	7
2.2.3	Zadání z pohledu QuickControl	7
2.3	Stávající možnosti řízení	7
2.3.1	Modul GUI	8
2.3.2	Modul HTTP server	8
2.4	Dostupné konkurenční produkty	8
2.4.1	Control Web	9
2.4.2	Simatic Thin client	9
2.4.3	Web Navigator	9
2.5	Výsledky analýzy	10
3	Specifikace požadavků	11
3.1	Funkční požadavky	11
3.2	Spolehlivost	12
3.3	Použitelnost, snadnost použití	13
3.4	Efektivnost	14
3.5	Udržovatelnost	15
3.6	Přenositelnost	16
3.7	Konkrétní požadavky	16
4	Výběr technologie	17
4.1	Hardware	17
4.1.1	Technické parametry	17
4.2	Operační systém	18
4.2.1	Možnosti uživatelského softwaru	18
4.3	Software	20
4.3.1	Řešení s použitím stávajícího HTTP serveru	20
4.3.2	Řešení s nutností implementace serveru	22
4.4	Vybrané řešení	23

5	Použitý software	24
5.1	Xvnc	24
5.1.1	RFB protokol	25
5.2	Xinetd	27
5.2.1	Princip	27
5.2.2	Výhody	28
5.2.3	Konfigurace	29
5.3	IceWM – X window manager	30
5.4	QuickControl	30
5.5	Nástroje pro implementaci klienta	30
6	Návrh systému	31
6.1	Schéma systému	31
6.1.1	Instance serveru	32
6.1.2	Xinetd	32
6.1.3	qcvncshell.sh	32
6.1.4	Časový průběh spojení	33
6.2	Způsob konfigurace	33
6.2.1	Instalace softwaru	34
6.2.2	Konfigurace QuickControl	34
6.2.3	Konfigurace Xvnc	35
6.2.4	Konfigurace Xinetd	35
6.2.5	Konfigurace window manageru	36
6.2.6	Konfigurace skriptu qcvncshell.sh	36
7	Implementace	37
7.1	Server	37
7.1.1	Konfigurace Xinetd	37
7.1.2	Konfigurace IceWM	38
7.1.3	qcvncshell.sh	38
7.2	Klient	39
7.2.1	Windows Mobile klient	39
7.2.2	Java klient	40
8	Závěr	41

Kapitola 1

Úvod

Neustálý vývoj v oblasti miniaturizace elektroniky umožnil v posledních několika letech masové rozšíření malých přenosných zařízení s poměrně vysokým výkonem. V kapse dnes běžně lidé nosí zařízení s výpočetním výkonem přesahujícím možnosti stolních počítačů před deseti lety. Do budoucna lze nadále předpokládat zvyšování výkonu spolu se snižováním velikosti a spotřeby energie přenosných zařízení. To jsou fakta, která určují jeden ze současných trendů ve vývoji softwarových řešení. Vzniká čím dál více požadavků – více či méně opodstatněných – přizpůsobit funkčnost stávajících systémů pro běh na přenosném zařízení. Tento trend je patrný jak v oblasti elektroniky pro osobní použití, tak v oblasti průmyslu.

Zadání této diplomové práce vzniklo ve spolupráci s firmou ASYC s. r. o., která se zabývá vývojem, návrhem a realizací systémů měření a regulace. Jedná se o automatizaci řízení technologických procesů, převážně z oblasti energetiky, chemie, ekologie, vodohospodářství, dopravy, apod.

Také firma ASYC byla postavena před problémem, jak umožnit svým zákazníkům používat dodávaná softwarová řešení vzdáleně – pomocí přenosných zařízení. Nabídka vzdálené obsluhy systému může být totiž pro zákazníka rozhodujícím faktorem při výběru dodavatele. Bude-li moci firma ASYC vzdálenou obsluhu zákazníkům nabídnout, může to pro ni znamenat rozhodující krok k předstihu konkurenčních dodavatelů. Absence takové nabídky se naopak může stát důvodem oslabení pozice firmy ASYC na trhu automatizačních systémů.

Tato diplomová práce si klade za cíl poskytnout firmě ASYC univerzální konfigurovatelný nástroj umožňující obohatit stávající systémy měření a regulace o možnost vzdáleného řízení.

Kapitola 2

Současný stav

Návrhu nového softwarového řešení předchází analýza řešení současného s cílem identifikovat ty části stávajícího softwaru, které mohou být použity pro následnou implementaci. Také jsou porovnány vlastnosti obdobných konkurenčních produktů za účelem zpřesnění požadavků na řešení.

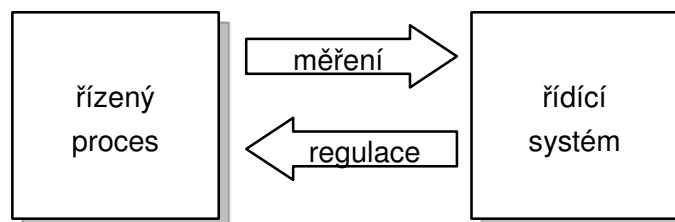
Ještě před tím však bude stručně nastíněna problematika systémů měření a regulace. Vývoj softwaru pro oblast měření a regulace je totiž vzhledem k ostatním oblastem informačních technologií v některých ohledech specifický. Tato specifika se promítají především do požadavků na řešení a je nutno je zohlednit jak při návrhu, tak při následné implementaci systému.

2.1 Systémy měření a regulace

Systémy měření a regulace nás v různých podobách obklopují téměř na každém kroku. Nacházejí uplatnění jak v průmyslu (elektrárny, čističky odpadních vod, hutnictví, aj.), tak čím dál více v domácnostech (regulace vytápění, vzduchotechnika, aj.). Princip činnosti je však stejný v průmyslu i v domácnosti.

2.1.1 Schéma systémů měření a regulace

Základní princip systému pro měření a regulaci je jednoduchý, sestává z *řízeného procesu* a *řídícího systému*. Mezi řízeným procesem a řídicím systémem probíhá obousměrná komunikace. Blokové schéma je uvedeno na obrázku 2.1.

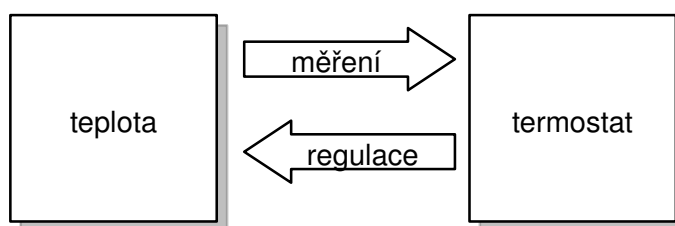


Obrázek 2.1: Základní schéma řízení procesu.

Vysvětlení pojmů ze schématu 2.1 je následující:

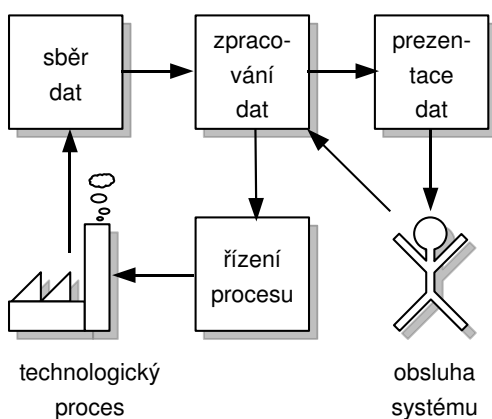
- **Řízený proces** je technologický či výrobní proces, sledovaná fyzikální veličina, apod.
- **Řídicí systém** je takový systém, který dokáže v čase měnit průběh řízeného procesu. Součástí systému je také jeho obsluha.
- **Měření** znamená získávání dat z řízeného procesu.
- **Regulací** rozumíme změnu průběhu procesu.

Jednoduchým příkladem systému měření a regulace může být například vytápění rodinného domku regulované termostatem (obr. 2.2). Řízeným procesem je teplota v domě, řídicím systémem je jednoduchý termostat. Měřením je získávána aktuální teplota, regulací se rozumí zapnutí nebo vypnutí vytápění.

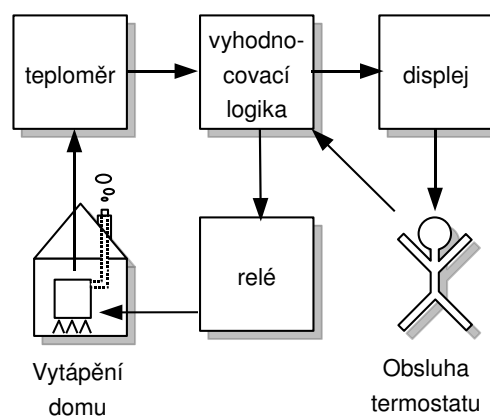


Obrázek 2.2: Schéma systému řízení teploty.

Blok řídicího systému popíšeme podrobněji. Podle oblasti nasazení, množství monitorovaných dat a dalších specifických požadavků se od sebe různé řídicí systémy v jistých ohledech liší, principiální schéma je však vždy stejné. Každý systém měření a regulace lze zjednodušeně popsat blokovým schématem vyobrazeným na obrázku 2.3. Na obrázku 2.4 je pak totéž schéma pro příklad vytápění s termostatem.



Obrázek 2.3: Podrobnější schéma řídicího systému.



Obrázek 2.4: Podrobnější schéma systému řízení teploty.

Jednotlivé bloky systému řízení z obrázku 2.3 budou blíže popsány v následujících odstavcích.

Sběr dat

Tato část systému měření a regulace zajišťuje zásobování systému aktuálními daty z regulovaného procesu. Sběr dat je realizován různými typy senzorů (teploměry, tlakoměry, apod.). O tom, která data jsou pro řízení procesu důležitá a jak často je potřeba data z procesu získávat rozhoduje návrhář systému podle specifikace požadavků a dynamiky řízeného procesu.

V příkladu vytápění s termostatem se sběrem dat rozumí získání informace o aktuální teplotě v místnosti.

Zpracování dat

Zpracováním dat se rozumí souhrn algoritmů a postupů, pomocí kterých systém z dílčích dat získává znalosti, na jejichž základě je možno následně proces řídit.

V příkladu s vytápěním je zpracováním dat prosté porovnání aktuální teploty s teplotním maximem nastaveným na termostatu.

Prezentace dat

Prezentace dat umožňuje obsluze systému okamžitý přehled o stavu systému a stavu řízeného procesu. O tom, která data zobrazit, rozhoduje návrhář systému. Účelem je abstrahovat od nedůležitých údajů a zdůraznit ty důležité. Zobrazovat lze buď přímo číselnou hodnotu formou textu, nebo použít grafický způsob zobrazení (posuvník, graf, apod.). Tato část může v určitých případech v systému řízení zcela chybět.

V příkladu vytápění s termostatem by se prezentací dat rozumělo například zobrazení aktuální teploty (stav procesu), zobrazení nastavené maximální hodnoty a indikátoru zapnutého či vypnutého ohřevu (stav systému).

Řízení procesu

Blok řízení procesu mění aktuální stav procesu. Sestává z algoritmů a akčních členů. Na základě algoritmů vytvořených návrhářem systému jsou vyhodnocena zpracovaná data a následně modifikován stav akčních členů ve smyslu dosažení požadavků na funkčnost systému. Akčními členy mohou být ventily, klapky, relé, apod.

V příkladu vytápění s termostatem je akčním členem relé spínající topení. Algoritmus je velmi jednoduchý:

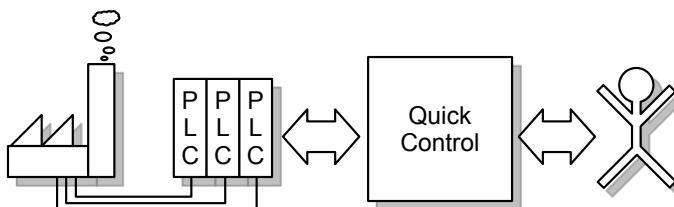
```
pokud teplota > nastavené maximum
    vypni topení
jinak
    zapni topení
```

2.2 QuickControl

Externí zadavatel této práce - firma ASYC s. r. o. dlouhodobě pracuje na vývoji vlastního softwaru pro systémy měření a regulace se zaměřením na rozsáhlé aplikace s velkým množstvím dat. QuickControl je v současnosti určen pro platformu PC a je vyvíjen pro operační systémy Windows a Linux.

2.2.1 Vztah ke schématu systémů měření a regulace

QuickControl plně pokrývá blok zpracování dat, blok prezentace dat obsluhuje a blok řízení procesu. Sběr dat a řízení akčních členů probíhá v součinnosti s programovatelnými logickými automaty, tzv. PLC¹. Blokové schéma systému měření a regulace s využitím softwaru QuickControl je na obrázku 2.5.



Obrázek 2.5: Blokové schéma řízení procesu pomocí QuickControl.

2.2.2 Klíčové vlastnosti QuickControl

QuickControl je plně konfigurovatelný, objektově orientovaný, modulární software, jehož jednotlivé moduly se starají o dílčí úkoly potřebné pro realizaci požadované funkčnosti řídicího systému (např. zpracování dat z PLC, ukládání dat do SQL databáze, zobrazení trendů, komunikace přes sériovou linku, apod.). Realizace konkrétní zakázky pak ze softwarového hlediska v ideálním případě sestává z výběru a konfigurace vhodných modulů.

Konfigurace aplikace v QuickControl je zajištěna soubory ve formátu XML². Součástí softwaru QuickControl je i editor, pomocí něž je možno tyto soubory snadno a efektivně editovat a měnit tak konfiguraci aplikace podle požadavků.

2.2.3 Zadání z pohledu QuickControl

Z hlediska externího zadavatele – firmy ASYC – zadání této práce zní:

Vytvořte samostatný systém vzdáleného řízení, který ve spojení s aplikací QuickControl umožní uživateli vzdálený monitoring a řízení technologického procesu pomocí přenosného zařízení. Systém musí být samostatně plně konfigurovatelný a musí vhodným způsobem zobrazovat objekty QuickControl reprezentující data technologického procesu. Systém vzdáleného řízení bude pracovat v režimu „klient - server“. Klientská část bude data prezentovat obsluhuje a poběží na přenosném zařízení (mobilní telefon, PDA³). Klientský software by měl pracovat na co nejširší škále přenosných zařízení.

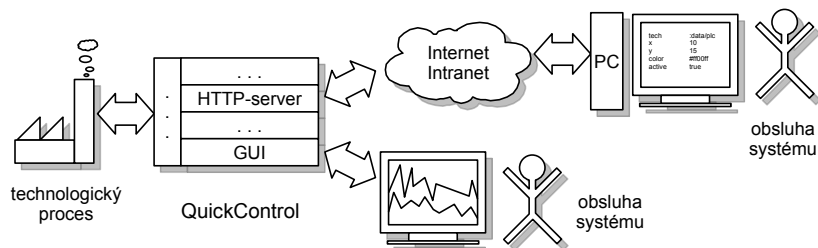
2.3 Stávající možnosti řízení

QuickControl v současné době umožňuje dva způsoby řízení technologického procesu, každému z nich odpovídá jeden konfigurovatelný modul. Obě dvě možnosti jsou graficky znázorněny na schematicém obrázku 2.6 a budou popsány dále.

¹PLC je zkratka pro Programmable Logic Controller (programovatelný logický automat).[18]

²XML je zkratka pro Extensible Markup Language, značkovací jazyk vyvinutý konsorciem W3C.

³PDA je zkratka pro Personal Digital Assistant (osobní digitální asistent).[17]



Obrázek 2.6: Možnosti řízení procesu systémem QuickControl.

2.3.1 Modul GUI

Modul GUI je určen k prezentaci monitorovaných dat lokálně na monitoru počítače, na němž běží systém QuickControl. V konfiguračním souboru pro modul GUI je možno nadefinovat libovolnou sadu ovládacích prvků spolu s daty, která se mají zobrazovat. Kromě standardních ovládacích prvků (tlačítka, záložky, posuvníky, apod.) nabízí použití speciálních vizualizačních komponent, jakými jsou například grafy trendů, výstrahy systému, aj.

Z hlediska systému vzdáleného řízení představuje modul GUI referenční model, podle něž bude posuzována použitelnost systému vzdáleného řízení. Vše, co je možno zobrazit obsluze systému pomocí modulu GUI na monitoru osobního počítače, musí být možno stejným či podobným způsobem zobrazit na obrazovce přenosného zařízení.

2.3.2 Modul HTTP server

Modul HTTP server umožňuje systému QuickControl poskytovat technologická data vzdálenému klientovi pomocí standardního protokolu HTTP⁴. Umožňuje rovněž přijímat od klienta požadavky na změnu a tím technologický proces řídit. HTTP klientem je nejčastěji webový prohlížeč, v němž jsou pomocí jazyka HTML⁵ data zobrazena a následně asynchronním JavaScriptem⁶ pravidelně aktualizována. Tento modul de-facto umožňuje úplnou kontrolu nad řízeným procesem. Jeho hlavním nedostatkem je především velmi omezené grafické uživatelské rozhraní bez možnosti zobrazit speciální vizualizační prvky (trendy, alarmy, apod.). V řešení systému vzdáleného řízení je možno HTTP server vhodně upravit a použít tak, aby bylo dosaženo požadované funkčnosti.

2.4 Dostupné konkurenční produkty

V oblasti měření a regulace působí mnoho, větších či menších, společností. Některé z nich již svým zákazníkům možnost vzdáleného řízení nabízejí. Může být účelné zjistit klíčové vlastnosti a případné nedostatky těchto produktů a poučit se z nich pro následný návrh a implementaci.

⁴HTTP je zkratka pro Hypertext Transfer Protocol. Označuje protokol určený pro přenos souborů ve formátu HTML.[12]

⁵HTML je zkratka pro Hypertext Markup Language. Označuje značkovací jazyk, standard pro publikování dokumentů na internetu.[11]

⁶ JavaScript je interpretovaný, objektově orientovaný programovací jazyk používaný při publikování na internetu.

2.4.1 Control Web

Control Web je produkt společnosti Moravské přístroje, a.s. Jako svou součást obsahuje konfigurovatelný HTTP server, díky němuž umožňuje vzdálené řízení za pomoci standardního webového prohlížeče. Demo-aplikace je online přístupná na internetových stránkách společnosti⁷.

Klíčové vlastnosti

- Konfigurovatelnost podle technického vybavení klienta (čisté HTML, DHTML⁸, Java applety⁹, ActiveX¹⁰).
- Není nutno instalovat speciální klientský software.

Nedostatky

- Standardní HTML je statické, je nutno opakovaně žádat o data ze serveru. Pokročilejší webové technologie (AJAX¹¹, ActiveX, Java Applety) nejsou obvykle ve webových prohlížečích mobilních zařízení dostupné.

2.4.2 Simatic Thin client

Simatic Thin client dodávaný společností Siemens je technologie související s použitím dotykových displejů v automatizaci. Je postavena na protokolu RDP¹².

Klíčové vlastnosti

- Standardní protokol, existence klientů na mnoha platformách.
- Možnost integrace libovolného systému třetích stran běžícího na OS Windows.

Nedostatky

- Omezeno na systémy Windows.

2.4.3 Web Navigator

Web Navigator je součástí produktu WinCC od společnosti Siemens. Svými vlastnostmi je určen především pro použití jako tenký klient plnohodnotného PC než pro mobilní zařízení.

⁷<http://cw.mii.cz/>

⁸DHTML je zkratkou pro Dynamické HTML a označuje soubor technik k dosažení stránek interagujících s uživatelem.^[5]

⁹Java applet je program napsaný v jazyce Java určený pro běh v okně webového prohlížeče.

¹⁰ActiveX je technologie společnosti Microsoft umožňující běh aplikací v okně webového prohlížeče.

¹¹AJAX je zkratkou pro *Asynchronous JavaScript and XML* a označuje soubor technologií používaných pro dynamicky měnící se webové stránky bez nutnosti načítání celé stránky.^[2]

¹²RDP je zkratkou pro Remote Desktop Protocol používaný v operačních systémech společnosti Microsoft pro připojení vzdálené plochy.^[19]

Klíčové vlastnosti

- Umožňuje na klientu zřídit tzv. *dedikovaný web server*, kdy se klient stane serverem pro další klienty. Tím lze zajistit distribuci zátěže ve velké síti.
- Používá technologii ActiveX.
- Až 50 klientů.

Nedostatky

- Omezení na systémy Windows.
- Příliš velké nároky na softwarové vybavení klienta (Internet Explorer 6, ActiveX).

2.5 Výsledky analýzy

V současné době se k řízení technologického procesu ve společnosti ASYC používá software QuickControl. Ten již obsahuje HTTP server, jehož vhodným rozšířením by bylo možno dosáhnout obdobné funkčnosti, jakou poskytuje například konkurenční produkt Control Web. Vhodnost takového řešení bude dále diskutována v kapitole 3 - specifikace požadavků a v kapitole 4 - výběr technologie.

Konkurenční produkty společnosti Siemens jsou svou implementací orientovány pouze na platformu Windows. Velmi účelný se však jeví koncept integrace softwaru třetích stran, využitelný dobře v situaci, kdy je starý řídicí systém postupně nahrazován novým.

Kapitola 3

Specifikace požadavků

Každý softwarový produkt lze hodnotit podle mnoha kvalitativních charakteristik, které přímo ovlivňují jeho komerční úspěšnost. V každé oblasti, v níž se informační technologie ve větší míře uplatňují jsou obecně různé požadavky na jednotlivá kvalitativní kritéria. Zde je uvedena specifikace požadavků na kvalitu systému podle metrik uvedených v normě ČSN ISO/IEC 9126-1¹. Je přihlédnuto k použití v oblasti automatizace řízení.

3.1 Funkční požadavky

Účelnost

Účelnost je základním kritériem hodnocení kvality. Každý software musí především splňovat účel, pro nějž byl navržen. Systém vzdáleného řízení musí umožňovat dvě základní funkce:

- monitoring technologického procesu
- řízení technologického procesu

Přesnost, správnost

U systémů měření a regulace je velmi důležité, aby stav řídicího systému co nejpřesněji odpovídal stavu řízeného procesu. Jinými slovy, aby to, co vidí obsluha systému na monitoru, odpovídalo realitě technologického procesu. Tuto kvalitu do značné míry ovlivňuje návrhář systému měření a regulace. Použitý software by mu v tomto směru neměl klást žádná výrazná omezení (například počet zobrazených desetinných míst, rozmístění vizualizačních prvků, apod.).

Kompatibilita

Serverová část systému musí být kompatibilní se serverovým operačním systémem. Vybrané přenosné zařízení, na němž bude provozována klientská aplikace pak musí být kompatibilní s ostatními prvky v síti, v níž bude systém provozován. Tato kompatibilita bude zajištěna na úrovni komunikačního protokolu.

¹ČSN ISO/IEC 9126-1 – Softwarové inženýrství – Jakost produktu – Část 1: Model jakosti

Shoda

V oblasti měření a regulace existuje celá řada závazných norem, nezávazných doporučení a zažitých konvencí. Tato pravidla by měl každý systém pro měření a regulaci respektovat. Do značné míry tuto vlastnost ovlivňuje návrhář systému, použitý software by mu v tomto směru neměl klást žádné překážky. Referenčním modelem pro systém vzdáleného řízení je modul GUI ze systému QuickControl.

Bezpečnost

Při řízení technologického procesu vzniká často požadavek ochrany citlivých dat před neautorizovaným přístupem. Získání technologických dat konkurencí může znamenat finanční ztráty či ztráty pozice na trhu. V některých oblastech nasazení je nutno přemýšlet o bezpečnosti v kontextu možných teroristických útoků. (jaderná energetika, chemie, apod.). Tam je otázka bezpečnosti jedním z nejdůležitějších faktorů rozhodujících o výběru řídicího systému.

Otázka bezpečnosti může být řešena na různých úrovních (personální, fyzická, informační technologie, apod.), ve většině případů se kombinuje více přístupů k zabezpečení systému před neautorizovaným přístupem. Systém vzdáleného řízení by měl proto volitelně umožňovat dostatečnou úroveň zabezpečení jak na straně serveru (autentizace klienta před poskytnutím dat), tak na straně klienta (heslo uživatele pro obsluhu). V ideálním případě by měl systém vzdáleného řízení umožňovat ochranu dat i na cestě mezi serverem a klientem (šifrování).

3.2 Spolehlivost

Spolehlivost je v oblasti měření a regulace klíčová vlastnost. Ukáže-li se, že systém není spolehlivý a selhává, klesá důvěra obsluhy v systém, klesá kredit dodavatelské firmy. Selhání jednoho modulu může vést k selhání celého systému, selhání systému vede často k nemalým finančním ztrátám.

Zralost

Zralost je vlastnost softwarového produktu vyjadřující četnost selhání softwaru v důsledku chyb v softwaru. Ta by se v případě systému vzdáleného řízení měla limitně blížit nule. Chyb v softwaru se samozřejmě vyvarovat nelze, ale je nutno vyvinout maximální úsilí k minimalizaci rizika úplného selhání systému, především jeho serverové části.

Odolnost vůči chybám

V procesu získávání a zpracování technologických dat může na mnoha místech docházet k chybám. Důvodem může být chyba v softwaru, hardwarová porucha nebo chyba lidského faktoru. To se může projevit například ztrátou spojení mezi klientem a serverem, neočekávaným formátem dat či jejich úplnou absencí. Ve všech případech by měl být software schopen se s chybami vyrovnat a vhodným způsobem problém prezentovat obsluze.

Schopnost zotavení

V případě, že dojde v systému měření a regulace k chybě, je vždy cílem co nejrychleji uvést systém do funkčního stavu za účelem minimalizace rizika ztrát. Software by měl být tedy

navržen tak, aby ihned po odstranění závady (výměna vadného senzoru, zapnutí přenosného zařízení, apod.) automaticky navázal spojení a pokračoval v předchozí činnosti bez nutnosti zásahu obsluhy.

3.3 Použitelnost, snadnost použití

Často při nasazování automatizace do oblasti řízení technologického procesu nastává situace, kdy je starý systém řízení nahrazován novým, moderním. Co však ve většině případů zůstává nezměněno jsou uživatelé systému, tedy jeho obsluha. Pro tu je důležitou vlastností systému snadnost jeho použití.

Pochopitelnost

Pochopitelnost softwarového produktu určuje, jak rychle je schopen uživatel pochopit princip jeho ovládání. V kontextu systému vzdáleného řízení se jedná hlavně o pochopitelnost ovládání vybraného přenosného zařízení. Ovládání konkrétní aplikace by mělo být stejné jako ovládání pomocí modulu GUI stávajícího systému QuickControl a je závislé na návrháři aplikace.

Ovladatelnost

Ovladatelnost aplikace úzce souvisí s uživatelským rozhraním. V případě aplikací určených pro přenosné zařízení je uživatelské rozhraní velmi specifické a odlišné od uživatelského rozhraní aplikace pro PC.

Uživatelské rozhraní je do značné míry determinováno množinou typů událostí, které je zařízení, na němž aplikace běží, schopno zpracovat. Úkolem aplikace je na události zařízením podporované adekvátně reagovat a události zařízením nepodporované vhodným způsobem nahradit. Cílem řešení je stav, kdy všechny prvky uživatelského rozhraní jsou uživateli jistým způsobem přístupny na všech přenosných zařízeních.

Referenčním modelem uživatelského rozhraní pro systém vzdáleného řízení je modul GUI. Ten rozpoznává a aktivně využívá následující typy událostí vstupu od uživatele:

- Stisknutí klávesy či stisknutí více kláves současně.
- Stisknutí levého tlačítka myši.
- Dvojitě stisknutí levého tlačítka myši.
- Stisknutí pravého tlačítka myši.

Zdaleka ne na všech typech přenosných zařízení jsou všechny tyto události dostupné. Například většina mobilních telefonů nemá žádné polohovací zařízení a nemá tedy prostředky pro generování událostí podobných stisknutí tlačítka myši. Rovněž vstup textu je u většiny mobilních telefonů specifický, sestává ze sekvence stisknutí stejné klávesy. Klientská část systému vzdáleného řízení proto musí umět transformovat například sekvenci tří stisknutí klávesy '4' na stisknutí klávesy 'i' na běžné klávesnici. Absenci polohovacího zařízení a modifikačních kláves (ctrl, alt, shift) je rovněž nutno nějakým způsobem vyřešit.

3.4 Efektivnost

Efektivností se rozumí výkon aplikace v poměru k použitým zdrojům. Při návrhu softwaru pro přenosné zařízení je otázka efektivnosti problémem, který nemůže být podceněn. Přestože výkon přenosných zařízení neustále roste, je stále řádově nižší, než u běžných osobních počítačů. Při návrhu klientské části systému vzdáleného řízení je třeba počítat s několika omezeními.

Omezený výpočetní výkon

Výrobci přenosných zařízení stojí vždy před problémem, jak maximalizovat výpočetní výkon přístroje při zachování limitujících faktorů. Těmi jsou hlavně cena, velikost zařízení a spotřeba elektrické energie. V současné době jsou přenosná zařízení osazována převážně tzv. ARM² procesory s RISC architekturou³ a taktovací frekvencí 100-1000 MHz.

Omezení velikosti paměti pro program

V některých zařízeních je omezená velikost spustitelné aplikace. Toto omezení může být velmi výrazné, například jen 64 KB u starších typů mobilních telefonů, většinou se ale pohybuje okolo 1 MB. Novější zařízení nemají pro velikost programu žádná omezení vyjma celkové velikosti dostupné paměti. Výsledná klientská část systému vzdáleného řízení by přesto neměla přesáhnout 1 MB.

Omezení velikosti paměti pro dynamickou alokaci

Téměř každý software potřebuje k vykonávání své činnosti za běhu alokovat paměť pro svá data. K tomu je v každém zařízení určena paměť pro dynamickou alokaci, tzv. halda. Ta může mít buď fixní velikost (typické pro mobilní telefony), nebo se může měnit podle aktuálního požadavku až do maximální velikosti dostupné paměti. V případě mobilních telefonů je velikost haldy 200 KB - 2 MB, u novějších typů telefonů a PDA již není velikost haldy omezena. Čím méně dat bude klientská aplikace dynamicky alokovat, tím více zařízení ji bude moci provozovat. Aby bylo zajištěno, že dynamická paměť nedojde, musí být garantována horní hranice alokované paměti, nebo musí být aplikace schopná se s nedostatkem paměti vyrovnat.

Omezení velikosti paměti pro data

Přestože vnitřní paměť pro data není u většiny přenosných zařízení příliš velká (500 KB - 16 MB), umožňují současné přístroje tuto paměť rozšířit pomocí paměťových karet různých typů. Ty mohou dosahovat velikostí až 16 GB. Velikost paměti pro data tedy nepředstavuje pro provoz aplikace výrazné omezení, přesto by měla být zajištěna horní hranice použité datové paměti.

Omezení šířky pásma připojení

U přenosných zařízení se předpokládá připojení k síti internet pomocí některé z technologií bezdrátového připojení. Mobilní telefony disponují v současnosti možností připojení pomocí

² ARM je zkratkou pro Advanced RISC Machine (pokročilý stroj s architekturou RISC).[3]

³ RISC je zkratkou pro Reduced Instruction Set Computer(počítač s redukovanou instrukční sadou).

HSCSD	5 KB/s
GPRS	10 KB/s
EDGE	30 KB/s
Bluetooth	90 KB/s
HSDPA	1,8 MB/s
Wi-Fi	6 MB/s

Tabulka 3.1: Hodnoty maximální propustnosti technologií bezdrátového připojení.

technologií HSCSD⁴, GPRS⁵, EDGE⁶, Bluetooth⁷, případně HSDPA⁸. U zařízení typu PDA bývá k dispozici nejčastěji připojení Bluetooth a Wi-Fi⁹. Rozhodující vlastností technologie bezdrátového připojení je maximální propustnost, tedy množství přenesené informace za jednotku času. Tabulka 3.1 ukazuje hodnoty maximální propustnosti pro jednotlivé technologie bezdrátového připojení.

Omezení velikosti zobrazovacího zařízení

Mobilní zařízení mají oproti osobním počítačům 4–10× menší displej. I s tímto faktem je potřeba při návrhu systému vzdáleného řízení počítat a poskytnout i na tak malé ploše obsluze dobrý přehled o řízeném procesu. Velmi pravděpodobně bude ale velká část zodpovědnosti za přehlednost ležet na návrháři vizualizace, který bude muset pro zobrazení na přenosném zařízení navrhnout speciální rozmístění vizualizačních prvků.

3.5 Udržovatelnost

Systém vzdáleného řízení není jednoúčelová aplikace, předpokládá se její dlouhodobé používání s možností úprav za účelem rozšíření, adaptace na nové prostředí či opravy chyb. Proto je velký důraz kladen na udržovatelnost. Všechny zdrojové kódy musejí být důsledně komentovány, součástí řešení musí být jak uživatelská, tak vývojářská dokumentace. Mimo to by měl systém dosahovat přiměřených hodnot kvality následujících charakteristik softwarových produktů.

Konfigurovatelnost

Již bylo zmíněno, že každý modul systému QuickControl je konfigurován pomocí konfiguračního souboru ve formátu XML. Součástí řešení tedy musí být rovněž návrh obsahu konfiguračního souboru za účelem plné konfigurovatelnosti všech podstatných částí systému.

⁴HSCSD je zkratka pro High-Speed Circuit-Switched Data (vysokorychlostní přenos dat přepínáním okruhů).[9]

⁵GPRS je zkratka pro General Packet Radio Service.[7]

⁶EDGE je zkratka pro Enhanced Data Rates for GSM Evolution, označuje tzv 2.75 generaci mobilních sítí.[6]

⁷Bluetooth je bezdrátová komunikační technologie sloužící k bezdrátovému propojení mezi dvěma a více elektronickými zařízeními.[32]

⁸HSDPA je zkratka pro High-Speed Downlink Packet Access. Protokol 3 generace mobilní telefonie.[10]

⁹Wi-Fi je bezdrátová komunikační technologie podle protokolu IEEE 802.11[23]

Testovatelnost

V oblasti vývoje softwaru pro měření a regulaci je obzvláště důležité, aby byl software před nasazením do reálného prostředí důkladně otestován. Systém vzdáleného řízení by měl být připraven na systematické otestování jednotlivých jeho částí.

Analyzovatelnost

Aby bylo možno v případě výskytu chyby softwaru nebo jiné poruchy systému zajistit co nejrychlejší nápravu, je nutné, aby software umožňoval analyzovat příčinu poruchy.

Modifikovatelnost

Systém vzdáleného řízení by měl být navržen dostatečně obecně, aby v něm bylo možno s minimálními časovými nároky a minimálním úsilím realizovat funkční změny a opravovat chyby. Vzhledem k faktu, že místo nasazení systému bývá často velmi vzdálené od místa sídla firmy realizující řídicí systém, je vhodné, aby bylo možno novou verzi softwaru aktualizovat vzdáleně – po síti.

3.6 Přenositelnost

Přenositelnost softwaru lze vyjádřit mírou vynaloženého úsilí nutného k přechodu na jiné systémové prostředí. V kontextu systému vzdáleného řízení je nutno nahlížet na přenositelnost ze dvou pohledů – přenositelnost klienta a přenositelnost serveru.

Přenositelnost serveru

Serverová část systému vzdáleného řízení poběží na některém z Linuxových operačních systémů. Jeho přenos na jiný typ OS se nepředpokládá.

Přenositelnost klienta

Uživatelský software v přenosném zařízení může buď běžet přímo na operačním systému daného zařízení, nebo využívat podpory virtuálního stroje či interpretu (JVM¹⁰, CLR¹¹, aj.). Cílem návrhu systému je, aby klient mohl běžet na co nejširší škále mobilních zařízení s minimálním úsilím při přenesení na jiné zařízení.

3.7 Konkrétní požadavky

Kromě požadavků na kvalitu výše uvedených kvalitativních charakteristik byly zadány externím zadavatelem tři explicitní požadavky na realizaci systému:

1. Implementovat klienta na zařízení PDA Acer n300 s operačním systémem Windows Mobile 5.0.
2. Implementovat klienta pro mobilní telefony s podporou technologie Java.
3. Vytvořit ukázkovou aplikaci běžící na těchto dvou klientech pro účely demonstrace funkčnosti zákazníkovi.

¹⁰ JVM je zkratka pro Java Virtual Machine.

¹¹ CLR je zkratka pro Common Language Runtime - součást technologie .NET Compact.

Kapitola 4

Výběr technologie

Jakmile je známa specifikace požadavků, které musí systém vzdáleného řízení splňovat, je možno přistoupit k výběru informačních technologií. Ty by měly umožnit řešení problému za pokud možno co nejnižších nákladů. Výběr informačních technologií zahrnuje například volbu operačního systému mobilního zařízení, volbu programovacího jazyka, dostupného vývojového prostředí, apod.

4.1 Hardware

Přenosná zařízení můžeme rozdělit na dva základní typy - mobilní telefony a osobní asistenti, tzv. PDA. Tyto dva typy zařízení se od sebe liší v několika aspektech. Na pomezí mezi mobilními telefony a PDA leží tzv. smartphony,¹ které rozdíl mezi nimi do jisté míry stírají a přebírají funkční vlastnosti obou. Lze předpokládat, že se technické parametry přenosných zařízení budou nadále vyvíjet a volba konkrétního zařízení bude vždy vedena aktuálními potřebami. Zde je tedy pouze uveden výčet orientačních hodnot parametrů současných zařízení.

4.1.1 Technické parametry

Technické parametry se u jednotlivých zařízení mohou výrazně lišit. Procesory (nejčastěji ARM, architektura RISC) mohou pracovat na frekvencích 100 – 1000 Hz, paměti může být k dispozici od 500 KB do 16 GB. Rozlišení displeje se pohybuje od 96×65 u nejstarších mobilních telefonů až po například 800×480 obrazových bodů u některých smartphonů. Fyzická velikost displeje je limitována velikostí zařízení, které musí mít kompaktní rozměry. Maximální velikost se proto pohybuje okolo 8×6 cm.

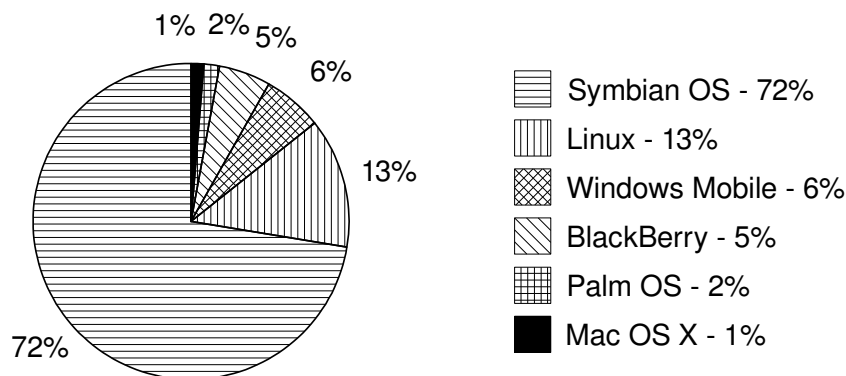
Pro bezdrátové připojení bývá u mobilních telefonů k dispozici HSCSD, GPRS, EDGE a Bluetooth, u PDA většinou Bluetooth a Wi-Fi, smartphone často umožňuje vše v jednom zařízení spolu s HSDPA. Ovládání je u mobilních telefonů řešeno pomocí numerické klávesnice spolu se směrovými tlačítky pro navigaci. PDA a smartphony poskytují dotykový displej a často i plnohodnotnou klávesnici. Pokud taková klávesnice k dispozici není, emuluje se na displeji.

K napájení se používají nejčastěji lithium-iontové baterie s kapacitou 800-1700 mAh. Výdrž baterie je nepřímo úměrná výkonu zařízení a pohybuje se od 3 h po 7 dní v závislosti na typu zařízení a na míře používání energeticky náročných funkcí přístroje.

¹Smartphone je mobilní telefon, který poskytuje uživateli pokročilé funkce, blížící se svou využitelností osobním počítačům.[20]

4.2 Operační systém

Na trhu s univerzálními operačními systémy pro přenosná zařízení existuje několik hlavních výrobců. Na obrázku 4.1 je graf jejich podílu na světovém trhu zařízení typu smartphone v druhém kvartálu roku 2007.²



Obrázek 4.1: Graf podílu na světovém trhu operačních systémů v zařízeních typu smartphone v druhém kvartálu roku 2007.

4.2.1 Možnosti uživatelského softwaru

Od typu operačního systému, který výrobce ve svém přístroji použil se odvíjí možnost provozovat na daném zařízení uživatelský software.

Zařízení s proprietárním operačním systémem

Operační systém většiny mobilních telefonů je tzv. *aplikačně specifický* – je zaměřen na konkrétní oblast použití, na použití v mobilních telefonech. Každý výrobce vyvíjí operační systém pro svá zařízení zvlášť, aplikační programové rozhraní (API³) není veřejné. Vývoj softwaru nativního pro tyto operační systémy proto není možný.

Vždy má ale výrobce možnost implementovat ve svém zařízení virtuální stroj (například JVM pro Java MIDlety⁴), pro který je možno software vyvíjet. To má navíc tu výhodu, že je takto vyvinutý software přenositelný na všechna mobilní zařízení, která daný virtuální stroj implementují. Obrázek 4.2 ukazuje schéma vrstev, podílejících se na běhu takto implementovaného softwaru.

Zařízení s univerzálním operačním systémem

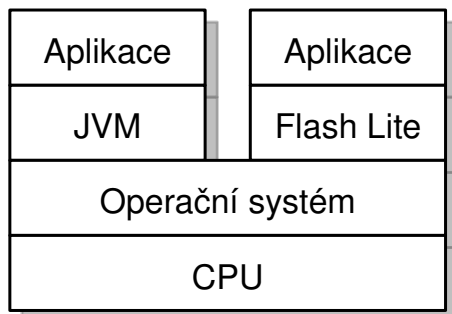
PDA a smartphony jsou téměř výhradně dodávány s nainstalovaným univerzálním operačním systémem. Aplikační rozhraní takových operačních systémů je známé a do určité míry shodné s verzí pro osobní počítače. Vývojář tak může vyvíjet software nativní pro daný

²Zdroj: Canalys, společnost zabývající se průzkumy v oblasti informačních technologií. <http://www.canalys.com>

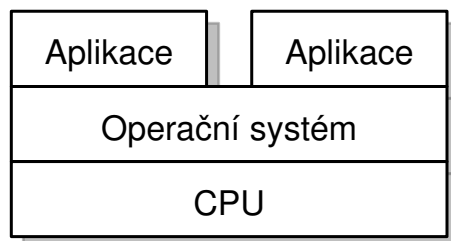
³API je zkratka pro Application Programming Interface (aplikační programové rozhraní). Je to rozhraní, které poskytuje operační systém nebo knihovny v něm obsažené.

⁴MIDlet je obdoba Java appletu určená pro přenosná zařízení.

operační systém. Schéma vrstev podílejících se na běhu nativního softwaru je na obrázku 4.3. Výhodou vývoje nativního softwaru je jeho vyšší efektivnost, nevýhodou nižší přenositelnost. Samozřejmě je možno i u těchto operačních systémů nainstalovat virtuální stroj a vyvíjet software podle schématu na obrázku 4.2.



Obrázek 4.2: Schéma aplikace běžící na virtuálním stroji.



Obrázek 4.3: Schéma aplikace nativní pro daný operační systém.

Symbian OS [21]

Operační systém dominantní mezi operačními systémy mobilních telefonů vyvíjený společností Symbian Ltd., ve vlastnictví předních výrobců přenosných zařízení (Nokia, Ericsson, Sony Ericsson, Panasonic, Siemens, Samsung). Symbian OS je určen výhradně pro procesory ARM. Pro vývoj aplikací pro Symbian OS je dostupná podrobná dokumentace k API, k dispozici jsou různá vývojová prostředí (tzv. SDK⁵) pro různé operační systémy. Vývoj nativních aplikací vyžaduje poměrně velké úsilí kvůli speciálním programovacím technikám (práce s pamětí, apod.). Symbian OS dále přímo podporuje běh Java aplikací – MIDletů.

Linux

Pod pojmem Linux se rozumí několik různých operačních systémů založených na Linu-xovém jádru. Používá jej například výrobce mobilních telefonů Motorola či sdružení firem Open Handset Alliance založené firmou Google za účelem vývoje operačního systému Android. Výhodou Linuxu je otevřenost zdrojového kódu, nevýhodou je prozatím nejednotnost v aplikačním rozhraní a neexistence standardu.

Windows Mobile [25]

Operační systém pro mobilní zařízení vlastněný a vyvíjený firmou Microsoft, setkat se lze rovněž s pojmem Pocket PC. Je dominantním na trhu operačních systémů pro PDA. K dispozici je opět SDK pro vývoj nativních aplikací v prostředí Microsoft Visual Studio. Windows Mobile rovněž přímo podporuje vývoj aplikací na platformě .NET Compact (blíže popsáno dále). Přímá podpora Javy ze strany Microsoftu není, existují ale implementace JVM pro Windows Mobile.

⁵SDK je zkratka pro Software Development Kit. Sada vývojových nástrojů pro danou platformu.

BlackBerry [4]

Pod názvem BlackBerry se skrývá přenosné zařízení a stejnojmenný operační systém, obě vyvíjené společností Research In Motion Limited (RIM). BlackBerry přímo podporuje Java MIDlety, ale jen ve verzi MIDP 1.0 (vysvětleno dále).

Palm OS [16] [29]

Palm OS, také znám jako Garnet OS, je vyvíjen společností Palm, Inc. Před několika lety zaznamenal velký nárůst podílu na trhu v oblasti zařízení typu PDA, v současné době je z trhu vytlačován operačním systémem Windows Mobile. Zdrojový kód Palm OS je uzavřený, existence dokumentace k API a SDK umožňuje vývoj nativního kódu pro Palm OS. Java není podporována přímo, existují ale implementace JVM pro Palm OS.

Mac OS X [14]

Mac OS X vyvíjí společnost Apple, Inc. pro svá přenosná zařízení. Je založen na unixovém jádru, má uzavřený zdrojový kód. Mac OS X umožňuje běh aplikací napsaných v Javě.

4.3 Software

Analýzou dostupných platforem a informačních technologií vzniklo několik návrhů na řešení zadaného problému. Je nutno vybrat takové řešení, které bude nejlépe splňovat specifikaci požadavků. Navrhovaná řešení lze rozdělit do dvou skupin. První skupina navrhovaných řešení obsahuje jako svou součást modul HTTP server a jejich hlavní implementační činností je tvorba klientské části. Druhá skupina řešení obsahuje ta řešení, která modul HTTP server nevyužijí a bude nutno pro ně vyvinout jak část klientskou, tak část serverovou.

4.3.1 Řešení s použitím stávajícího HTTP serveru

System QuickControl v současné době obsahuje modul poskytující technologická data pomocí protokolu HTTP. Tento modul neposkytuje dostatečný komfort používání kvůli omezenému grafickému uživatelskému rozhraní. Je však možné jej vhodným způsobem upravit či rozšířit a doplnit o klientskou část systému. Takové řešení má jednu výhodu – nižší náklady na vývoj serveru. O to složitější ovšem může být implementace klienta. Klient může být implementován různými způsoby, které jsou popsány dále.

Nativní software

Jednou z možností je vyvíjet klienta pro konkrétní operační systém vybraného přenosného zařízení. Výběr operačního systému může být proveden podle aktuální situace na trhu s operačními systémy pro mobilní zařízení.

Výhoda tohoto řešení je ve vyšší efektivitě prováděného kódu.

Nevýhod je hned několik, na prvním místě je velmi nízká přenositelnost, dále absence možnosti provozovat takový software na mobilním zařízení s proprietárním operačním systémem.

Pokud by cílem byla maximální efektivnost klienta, bylo by toto řešení vhodné. Pokud ale bude možno požadované úrovně výkonu dosáhnout jinými – přenositelnějšími – prostředky, dostanou pravděpodobně přednost.

HTML + JavaScript

Na velkém množství mobilních zařízení lze provozovat webový prohlížeč. Proto je jedním z navrhovaných řešení implementace pomocí standardních technologií pro vytváření WWW stránek, tedy použitím HTML, CSS⁶ a JavaScriptu.

Výhodou je usnadněná implementace – HTTP spojení a základní prvky GUI zajišťuje internetový prohlížeč. Dalšími výhodami je přenositelnost (i na osobní počítač) a snadná pochopitelnost pro uživatele.

Nevýhodou takového řešení je, že není zcela univerzální – je omezeno možnostmi internetového prohlížeče. Další nevýhodou je nízká efektivita kódu napsaného v JavaScriptu a nekompatibilita, případně absence funkcí JavaScriptu na různých přenosných zařízeních.

J2ME [13] [31]

Zkratkou J2ME se rozumí Java Platform, Micro Edition. J2ME je podmnožinou platformy Java vyvinutá s cílem poskytnout množinu aplikačního rozhraní pro aplikace běžící na mobilních zařízeních. Tato platforma je blíže specifikována takzvanými profily (Mobile Information Device Profile, MIDP), které určují konkrétní sadu použitých knihoven. Tyto profily se neustále vyvíjejí a s novými verzemi obsahují standard pro programování vyšších funkcí přenosného zařízení (Bluetooth, fotoaparát, apod.). U každého mobilního telefonu je uvedeno, jaký profil J2ME podporuje (MIDP 1.0, MIDP 2.0, JTWI⁷, MSA⁸, aj.).

Výhodou použití technologie J2ME je její současné veliké rozšíření mezi přenosnými zařízeními, převážně mobilními telefony, dále vysoká úroveň přenositelnosti díky běhu na virtuálním stroji, univerzálnost a rovněž existence hardwarové podpory u nových ARM procesorů (např. Jazelle⁹).

Nevýhodou je slabá podpora technologie Java v operačním systému Windows Mobile, současném lídru na trhu operačních systémů pro PDA. Je pravděpodobné, že společnost Microsoft bude prosazovat širší použití technologie .NET Compact na úkor J2ME. Další nevýhodou je relativně složitá implementace. Veškeré grafické uživatelské rozhraní bude muset být vybudováno od základů. Nabízí se možnost použití například standardní knihovny pro SVG¹⁰ (obrázek v SVG formátu je možno vygenerovat na serveru), to je ale obsaženo až ve specifikaci profilu MSA, který většina současných mobilních zařízení nepodporuje. J2ME starších profilů také neobsahuje podporu pro práci s čísly v plovoucí řádové čárce.

.NET Compact Framework [15] [33] [34]

Technologie .NET Compact Framework je podmnožinou platformy .NET Framework vyvíjené společností Microsoft za účelem vývoje software pro přenosná zařízení. Její součástí je virtuální stroj CLR - Common Language Runtime, na kterém běží aplikace přeložené do tzv. bajt-kódu. Principiálně je tato technologie obdobná technologii J2ME.

Výhodou .NET Compact je podpora na zařízeních s Windows Mobile, současném nejrozšířenějším operačním systémem v oblasti PDA.

⁶CSS je zkratkou pro Cascading Style Sheets (tabulky kaskádových stylů). Slouží k snadné úpravě vzhledu WWW stránek.

⁷JTWI je zkratkou pro Java Technology for Wireless Industry.

⁸MSA je zkratka pro Mobile Services Architecture, nejnovější standard mezi profily Java aplikací.

⁹Jazelle je technologie vyvinutá společností ARM Limited. Umožňuje novým ARM procesorům vykonávat přímo bajt-kód Javy.[3]

¹⁰SVG - Scalable Vector Graphics - Vektorový grafický formát.

Nevýhodou je současná slabá podpora na jiných zařízeních, převážně mobilních telefonech. Dále nízká přenositelnost, která je de-facto stejná jako při vývoji nativního softwaru pro Windows Mobile. Složitost implementace je obdobná jako u technologie J2ME.

Flash [1]

Technologie Flash byla původně vyvíjena společností Macromedia jako prostředek pro oživení WWW stránek vektorovou grafikou s možnostmi doprogramování funkčnosti pomocí tzv. ActionScriptu. Postupem času se ale z ActionScriptu stal plnohodnotný objektově orientovaný jazyk s možností samostatného překladu. K dispozici jsou funkce pro standardní požadované operace (HTTP spojení, zpracování událostí od uživatele, apod.). Pro běh na přenosném zařízení je zapotřebí přítomnost interpretu Flash Lite, funkční princip je obdobný jako u J2ME a .NET Compact Framework.

Výhodou technologie Flash je dobrá přenositelnost, Flash Lite je k dispozici na většinu operačních systémů, včetně operačních systémů pro osobních počítače. Další výhodou je vestavěná podpora formátu SVG.

Nevýhodou technologie Flash je nemožnost použití vyšších funkcí telefonu (Bluetooth, apod.), nízká podpora mezi staršími mobilními telefony a také nejistá budoucnost technologie v oblasti přenosných zařízení.

4.3.2 Řešení s nutností implementace serveru

Všechna předchozí navrhovaná řešení mají společné kromě použití HTTP serveru ještě to, že vykreslování vizualizace probíhá převážně na straně klienta na základě dat získaných po síti. To může být výhodné u jednoduchých grafických komponent (z jakých většinou vizualizace řízení technologických procesů složeny jsou), ale problematické u složitějších grafických prvků, například grafu trendů. V takovém případě by bylo zřejmě nutno na straně serveru vygenerovat bitovou mapu a přenášet ji po síti spolu s ostatními technologickými daty. Tak vznikla myšlenka vykreslování veškeré vizualizace na straně serveru a za použití vhodného protokolu přenášet po síti obrazová data. Samozřejmě musí být možno přenášet také uživatelské události vstupu od klienta k serveru.

VNC [22]

VNC je zkratka pro Virtual Network Computing a označuje technologii vzdáleného ovládání počítače pomocí sdílení pracovní plochy. VNC je založeno na RFB protokolu navrženém společností Olivetti & Oracle Research Lab. Umožňuje přenášet obrazová data ze serveru ke klientovi a události vstupních zařízení od klienta k serveru. Řešení by tedy sestávalo z implementace virtuálního VNC serveru, který by z technologických dat vytvářel vizualizaci podobně jako modul GUI s tím rozdílem, že výsledek by místo na monitor putoval po síti ke klientovi, který by obrazová data zobrazil.

Základní princip protokolu je jednoduchý. Obrazová data jsou přenášena od VNC serveru k VNC klientovi pomocí protokolu RFB. Server získává obrazová data od operačního systému, který na serveru běží, případně je nějakým způsobem vytváří sám. Takový VNC server se označuje jako virtuální. Existuje mnoho příkladů VNC serverů, které nezobrazují skutečnou pracovní plochu, ale generují například obrázky hodin podle stávajícího času, simulují ovládání přehrávače CD apod. Podobným způsobem může fungovat i virtuální VNC klient pro zobrazení technologického procesu v systému měření a regulace. Klient protokolu

RFB je skutečným tenkým klientem, nároky na jeho výkon jsou velmi nízké a může běžet na téměř libovolném zařízení, které má obrazovku.

Výhod řešení pomocí VNC je několik. VNC je široce rozšířen a téměř pro každý operační systém existuje nativní VNC klient, dostupná je také implementace v J2ME, která může běžet na mobilních telefonech s proprietárním operačním systémem. VNC klient je typickým příkladem tzv. tenkého klienta, který klade na výkon zařízení minimální nároky. Řešení pomocí VNC je robustní, obecné a velmi přenositelné.

Nevýhodou řešení pomocí VNC je relativně velká náročnost na šířku pásma připojení a nutnost implementace VNC serveru.

4.4 Vybrané řešení

Po pečlivém zvážení všech výhod a nevýhod všech navrhovaných řešení bylo rozhodnuto zvolit řešení pomocí technologie VNC. Rozhodujícími vlastnostmi byla robustnost, přenositelnost a obecnost.

Nevýhoda náročnosti na šířku pásma připojení není v aplikacích pro měření a regulaci tak markantní. Po většinu času je totiž obrazovka téměř neměnná, změny se v naprosté většině případů dějí lokálně (změna čísla, blikání kontrolky, apod.). Protokol RFB obsahuje několik možností, jak minimalizovat množství přenášených dat (komprese dat, přenos části zobrazované plochy, aj.), navíc umožňuje snadné rozšíření protokolu o vlastnosti specifické pro konkrétní nasazení.

VNC server nebude nutno implementovat, bude použito volně dostupných softwarových prostředků.

Kapitola 5

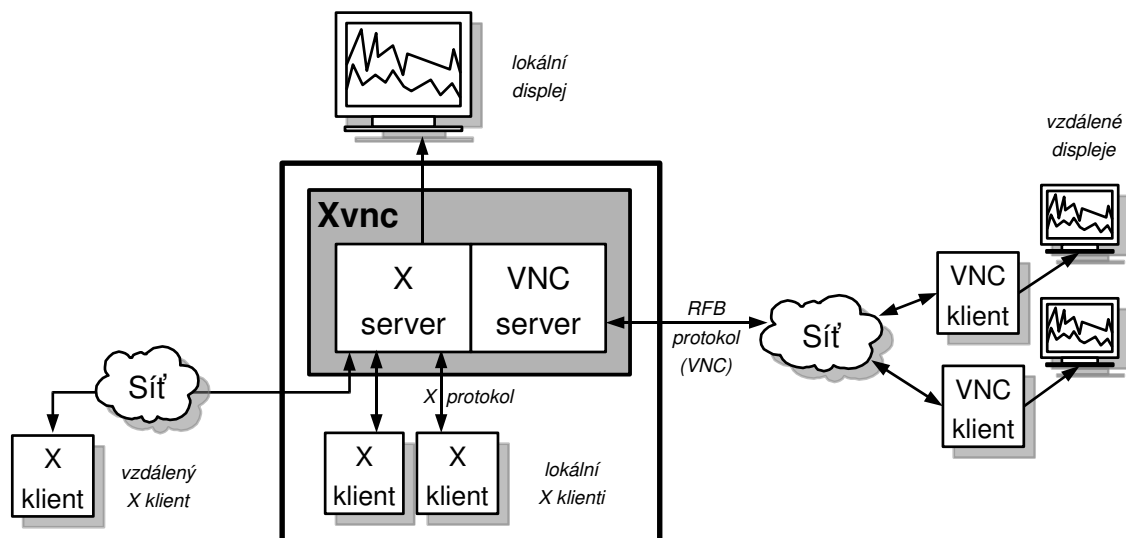
Použitý software

Volba vhodného softwaru může výrazně snížit množství nákladů potřebných pro implementaci systému. Zde jsou popsány jednotlivé softwarové nástroje, z nichž bude dále, v kapitole 6, vybudován výsledný systém vzdáleného řízení.

5.1 Xvnc

[26][27]

Xvnc je server vyvinutý společností RealVNC Ltd. Obsahuje funkcionalitu dvou serverů – X serveru a VNC serveru. X server je standardní unixový server, pomocí nějž mohou X klienti vykreslovat obrazová data na displej počítače. Ke komunikaci se používá *X window system core protocol*, zkráceně *protokol X*. Xvnc je z hlediska protokolu X standardní X server, jen místo na fyzickou obrazovku počítače vykresluje obraz do paměti, odkud jej pak VNC server přenáší RFB protokolem k VNC klientům. Blokové schéma sítě s počítačem vybaveným Xvnc serverem je na obrázku 5.1.



Obrázek 5.1: Schéma serveru Xvnc.

X server stojí mezi aplikacemi a hardwarem vstupu a výstupu a plní dvě základní funkce:

- Přijímá požadavky od X klientů na vykreslení obrazových dat na monitor. Klientem může být jakákoli aplikace s grafickým uživatelským rozhraním.
- Informuje X klienty o událostech. Jde jednak o události uživatelského vstupu (stisk klávesy, pohyb myši, apod.), jednak o události okna (zobrazení okna, posunutí okna, zvětšení okna, atd.).

Z hlediska požadavků na vykreslení na obrazovku se Xvnc chová stejně jako klasický X server, jen není nutně vázán na hardware stroje, na kterém běží. Ten totiž nemusí ani žádný monitor, klávesnici či myš mít. Místo na monitor vykresluje Xvnc data X klientů do paměti, kterou sdílí se serverem VNC. Odtud je při požadavku VNC klienta přenášén výsledný obraz po síti protokolem RFB (viz 5.1.1).

Události uživatelského vstupu přicházejí od VNC klientů k VNC serveru, který je dále poskytuje X serveru. Ten pak X protokolem informuje o událostech klienty. Z hlediska vnitřní architektury můžeme na jisté úrovni abstrakce o serveru XVNC mluvit jako o „překladači“ protokolu X na protokol VNC a naopak.

V systému vzdáleného řízení bude Xvnc server použit pro odesílání obrazových dat vizualizace po síti ke vzdáleným VNC klientům běžícím na přenosném zařízení. Podrobnosti budou rozebrány v kapitole 6 – návrh systému.

5.1.1 RFB protokol

RFB je zkratka slov Remote Frame-Buffer, tedy něco jako vzdálená paměť snímků. Jelikož protokol pracuje na úrovni framebufferu, může být použit pro jakýkoli operační systém či aplikaci. Jeho podrobnou specifikaci lze nalézt v [30].

Vlastnosti RFB protokolu

Základní vlastností RFB protokolu je jeho řízení klientem. Obrazová data nejsou kontinuálně posílána, jak je tomu například u protokolů pro přenos videa. Server pošle aktuální obsah framebufferu až na požadavek klienta. Tím je zajištěno, že klient stihne všechna data včas zpracovat. Další požadavek na obnovu framebufferu klient pošle až poté, co zpracuje odpověď serveru na předchozí požadavek. Zároveň je tím zajištěno, že obnova framebufferu probíhá tak rychle, jak to rychlost klienta a šířka pásma připojení dovolí. Pokud navíc k žádným změnám ve framebufferu serveru od minulého požadavku nedošlo, server žádná data neposílá až do doby, kdy ke změně dojde.

Další vlastnosti RFB protokolu je, že je bezstavový. Pokud se klient od serveru odpojí a znovu připojí, stav uživatelského rozhraní zůstává pro uživatele zachován.

Klientská strana protokolu je založena na základní operaci, kterou musí každý VNC klient umožňovat. Dala by se popsat jako „umísti obdélník obrazových dat na pozici X, Y“. Tento na první pohled neefektivní způsob vykreslování se stává poměrně efektivním tím, že RFB protokol definuje několik způsobů, jak tento obdélník obrazových dat popsat (kódování) a umožňuje rozšíření o další způsoby kódování obrazových dat. Obnovení aktuálního stavu framebufferu sestává ze sekvence obdélníkových obrazových dat.

RFB protokol poskytuje základní úroveň zabezpečení pomocí autentizace heslem. Heslo není po síti posíláno v textové formě, nicméně šifra sloužící k jeho zabezpečení je v reálném čase prolomitelná. Proto lze RFB protokol tunelovat například přes protokol SSH.

Průběh komunikace

Komunikace VNC klienta s VNC serverem pomocí protokolu RFB probíhá ve třech fázích. Tyto fáze jsou zobrazeny na obrázku [5.2](#)

- **Handshaking** – v této fázi se klient se serverem dohodnou na verzi protokolu a použitém zabezpečení.
- **Inicializace** – v této fázi server klientovi sdělí velikost framebufferu, barevný formát a jméno asociované se zařízením, na němž server běží.
- **Běžná interakce** – fáze, v níž klient může poslat libovolnou zprávu a obdržet na ni od serveru odpověď.

Klient – server zprávy

- **SetPixelFormat** – Klient požaduje nastavení barevného formátu pro obrazová data.
- **SetEncodings** – Klient sděluje serveru, které typy kódování umí zpracovat.
- **FramebufferUpdateRequest** – Klient požaduje obnovu specifikované oblasti obrazu.
- **KeyEvent** – Uživatel stiskl nebo pustil klávesu.
- **PointerEvent** – Uživatel pohnul myší, stiskl nebo pustil tlačítko myši.
- **ClientCutText** – Klient má nová textová data ve schránce.

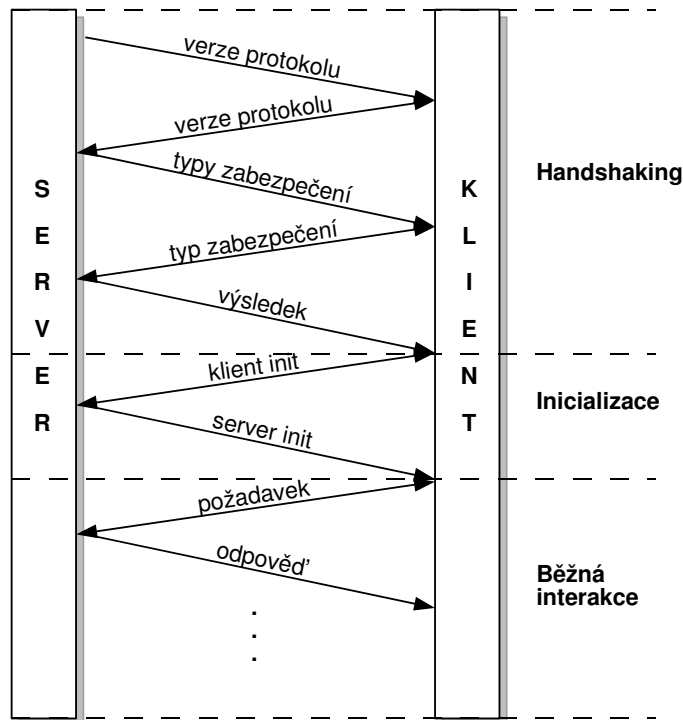
Server – klient zprávy

- **FramebufferUpdate** – Sekvence obdélníků obrazových dat. Odpověď na zprávu `FramebufferUpdateRequest` od klienta
- **SetColourMapEntries** – Pokud je použita paleta barev, sděluje touto zprávou server klientovi obsah tabulky barev.
- **Bell** – Požadavek na výstražné zvukové znamení, pokud to zařízení klienta umožňuje.
- **ServerCutText** – Server má nová textová data ve schránce.

Mechanismus rozšíření

Protokol RFB může být rozšířen ve třech různých oblastech:

- **Nová kódování** – Obrazová data mohou být zakódována libovolným novým způsobem. Pokud server klientem požadované kódování nepodporuje, prostě jej ignoruje. Klient také nikdy neobdrží data ve formátu, který nepodporuje, neboť si o takový typ kódování nikdy nezažádá.
- **Nová pseudo-kódování** – Pomocí pseudo-kódování lze od serveru vyžádat různá neobrazová data. Používá se například pro přenos souborů, přenos informací o změně velikosti framebufferu, apod.



Obrázek 5.2: Průběh komunikace mezi klientem a serverem pomocí protokolu RFB

- **Nové typy zabezpečení** – Protokol může být doplněn o nové typy zabezpečení. Tím je umožněno po fázi handshaking, kdy se server a klient dohodnou na typu zabezpečení, použít zcela jiný protokol pro komunikaci.

5.2 Xinetd

[28]

Xinetd je tzv. superserver, tedy server, který spravuje ostatní servery. S úspěchem je používán na unixových stanicích ke správě těch serverů, které nejsou v permanentním vytížení. Prvotní motivací k zavedení konceptu super-serveru pro poskytování síťových služeb bylo ušetření systémových prostředků serveru. Na jednom počítači totiž může běžet i několik desítek serverů mnoha různých síťových služeb. Běžně jsou v systému nainstalovány jako tzv. démoni¹ běžící na pozadí. Čekají na příchozí spojení, nevytěžují procesor serveru, ale zabírají část jeho operační paměti, a to i tehdy, když nejsou právě používány. V případě, že je takových démonů v systému velký počet a operační paměť serveru je malá, může díky tomu dojít k citelnému snížení výkonu serveru. Při řešení tohoto problému se zrodila myšlenka superserveru.

5.2.1 Princip

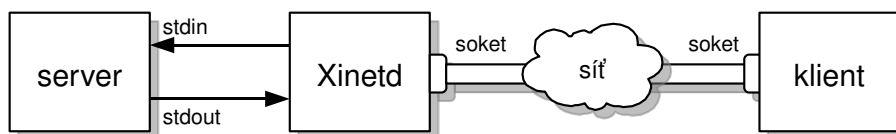
Základní princip superserveru je jednoduchý: Všechny démony síťových služeb nahradit jediným, který žádnou konkrétní službu neposkytuje, pouze spouští jednotlivé servery síťo-

¹Démon, orig. daemon, je pojem označující proces běžící na pozadí. Je obdobou pojmu služba z prostředí Windows. V Unixových systémech mají obvykle démoni jméno končící písmenem „d“ (sshd, xinetd apod.)

vých služeb podle aktuálního příchozího požadavku. Mezi takové superservery patří i Xinetd. Při startu Xinetd z konfiguračního souboru načte seznam všech služeb, které má spravovat.

V konfiguračním souboru jsou služby identifikovány podle jména (např. ftp, telnet). Číslo portu, na kterém služba běží může být uvedeno přímo v konfiguraci služby, často je však k překladu jmen na čísla portů použit seznam uvedený v souboru `/etc/services`. Přidělení čísel portů jednotlivým službám určuje organizace IANA² v dokumentu RFC 1340³ a je doporučeno je dodržovat. Pro použití služby, která není v RFC 1340 definována je vhodné zvolit takové číslo portu, které není obsazeno žádnou službou.

Jakmile má Xinetd k dispozici seznam portů, začíná poslouchat na síťovém rozhraní. Přejde-li paket s číslem portu některé ze spravovaných služeb, ověří nejprve Xinetd, zda je požadavek služby oprávněný. V konfiguračním souboru lze u každé služby definovat řadu podmínek, které musí příchozí spojení splňovat. Pokud je spojení v souladu s konfigurací služby, předá Xinetd spojení příslušnému serveru, který službu poskytuje. Spojení nemusí být serveru předáváno v podobě síťového soketu, Xinetd dokáže soket navázat na standardní vstup a standardní výstup serveru. To odstíňuje server od architektury sítě a umožňuje snadnou implementaci serveru bez nutnosti připojovat knihovny a psát kód pro práci se sítí. Server pouze čte ze standardního vstupu a zapisuje na standardní výstup. Tak může být server napsán například v některém jednoduchém skriptovacím jazyku. Schéma činnosti Xinetd je na obrázku 5.3.



Obrázek 5.3: Odstínění serveru od architektury sítě pomocí Xinetd.

5.2.2 Výhody

Přestože je na dnešních serverech s několika GB paměti problém vyčerpání paměti již překonán, je Xinetd nadále široce využíván pro ostatní výhody, které poskytuje. Některé z nich byly již zmíněny. Jedním z hlavních přínosů je sjednocený přístup k síťovým službám. Dále Xinetd poskytuje velmi dobrý mechanismus kontroly přístupu k jednotlivým službám, například podle denního času, IP adresy klienta, počtu spojení, apod. V neposlední řadě obsahuje Xinetd vestavěnou podporu obrany proti útokům typu DoS,⁴ spojení lze odmítnout či ukončit při vysoké zátěži serveru, příliš velkém počtu klientů z jedné IP adresy, příliš častému připojování ke službě, apod. Samozřejmostí je detailní zaznamenávání událostí do systémového log-souboru, případně do souboru definovaného konfigurací.

²IANA je zkratkou pro *Internet Assigned Numbers Authority* – organizaci, která přiděluje standardní čísla v oblasti internetových služeb.

³RFC je zkratkou pro *Request for Comments*. Pod tímto jménem jsou uveřejňovány dokumenty určující internetové standardy.

⁴DoS – Denial of Service – technika útoku na síťovou službu přehlcením služby požadavky

5.2.3 Konfigurace

Xinetd standardně načítá konfiguraci ze souboru `/etc/xinetd.conf`. V tomto souboru mohou být nadefinovány všechny služby, případně může být použito direktiv `include` a `includedir` pro definici služeb v externích souborech. Častým postupem je, že v souboru `/etc/xinetd.conf` je nakonfigurována pouze abstraktní služba `defaults`, která obsahuje implicitní nastavení pro všechny služby a direktivou `includedir /etc/xinetd.d` je připojen adresář, který obsahuje konfiguraci všech ostatních služeb.

Formát konfiguračního souboru

Konfigurace jedné síťové služby v konfiguračním souboru má následující formát:

```
service jméno_služby
{
    atribut_1 operátor hodnota
    ...
    atribut_N operátor hodnota
}
```

Operátory jsou tři:

- = přiřazuje atributu konkrétní hodnotu.
- += přidává hodnotu do výčtu hodnot.
- = odebírá hodnotu z výčtu hodnot.

Hodnotou může být číslo nebo řetězec, podle typu atributu. Jména atributů spolu s jejich významem budou uvedena dále.

Atributy konfigurace

Atributů v konfiguraci démona xinetd je mnoho a jsou detailně popsány v manuálových stránkách `xinetd.conf`. Zde je uveden soupis několika často používaných:

<code>server</code>	Cesta ke spustitelnému souboru serveru.
<code>server_args</code>	Parametry předané serveru na příkazové řádce.
<code>user</code>	Uživatelský účet, pod nímž se má služba spustit.
<code>port</code>	číslo portu služby (pokud není uvedeno v souboru <code>/etc/services</code>).
<code>instances</code>	Maximální počet běžících instancí serveru.
<code>per_source</code>	Maximální počet instancí spuštěných z jedné IP adresy.
<code>log_type</code>	Typ log-souboru (<code>SYSLOG</code> <code>FILE</code> [file]).
<code>log_on_failure</code>	Co zaznamenat do log-souboru při neúspěšném pokusu o spojení.
<code>log_on_success</code>	Co zaznamenat do log-souboru při úspěšném pokusu o spojení.
<code>only_from</code>	Rozsah IP adres, odkud je možno se ke službě připojit.
<code>no_acces</code>	Rozsah IP adres, odkud není možno se ke službě připojit.
<code>access_times</code>	Časový rozsah, kdy je možno se ke službě připojit.
<code>redirect</code>	Přesměrování spojení na jinou adresu v síti.

5.3 IceWM – X window manager

[24] IceWM je tzv. window manager, přesněji X window manager. Window manager je program, který kontroluje rozmístění a chování oken a řídí události uživatelského vstupu. X window manager je window manager pro protokol X. Z hlediska architektury X protokolu je klientem jako každá jiná aplikace, má však jisté specifické vlastnosti. Část interakce mezi X serverem a X klienty je přeměrována právě přes X window manager. Pokud se má například na obrazovce vykreslit nové okno (událost *expose*), X window manager určí pozici nově zobrazeného okna a postará se o vykreslení dekorací umožňujících uživateli manipulaci s oknem.

Pro implementaci systému vzdáleného řízení byl vybrán window manager IceWM, který byl svými tvůrci vyvinut s cílem poskytnout malý, rychlý a dobře ovladatelný window manager. IceWM klade důraz na rovnocennou ovladatelnost aplikací pomocí myši i klávesnice, což může být při návrhu vizualizace pro přenosné zařízení přínosem.

5.4 QuickControl

Pro řízení technologického procesu a pro vykreslování vizualizace bude použit software QuickControl popsáný v kapitole 2.2.

K návrhu vizualizace a ke konfiguraci modulů použitých pro řízení bude použit QCeditor, součást systému QuickControl.

5.5 Nástroje pro implementaci klienta

Použití standardního a široce rozšířeného protokolu RFB umožňuje minimalizaci nákladů při implementaci klientské části systému. Existuje řada VNC klientů pro různé platformy přenosných zařízení, šířených pod licencí GPL⁵. Licence GPL garantuje nejen bezplatné použití softwaru, ale i možnou modifikaci změnou zdrojového kódu. Modifikací je možno dosáhnout změny vzhledu (např. doplnění loga firmy) případně i změny funkčnosti, optimalizace výkonu, apod. Jedinou podmínkou plynoucí z licence GPL je uveřejnit takto implementovaného klienta opět pod licencí GPL.

Ve specifikaci požadavků v kapitole 3 je požadována implementace klienta pro OS Windows Mobile a klienta běžícího jako Java MIDlet. Těmto dvěma klientům odpovídají také dva vývojové nástroje:

- **Windows Embedded C++** bude použit pro implementaci klienta pro Windows Mobile. Je volně k dispozici ke stažení z webových stránek společnosti Microsoft.
- **Sun Java Wireless Toolkit** je nástroj vyvinutý společností SUN Microsystems, Inc. Slouží k překladu ze zdrojového kódu jazyka Java do Bajtkódu pro JVM a k vytváření Java MIDletů pro běh na mobilních zařízeních.

⁵GNU General Public License – Široce rozšířená licence pro svobodný software. Text licence je dostupný na webu GNU, <http://www.gnu.org/licenses/gpl.html>

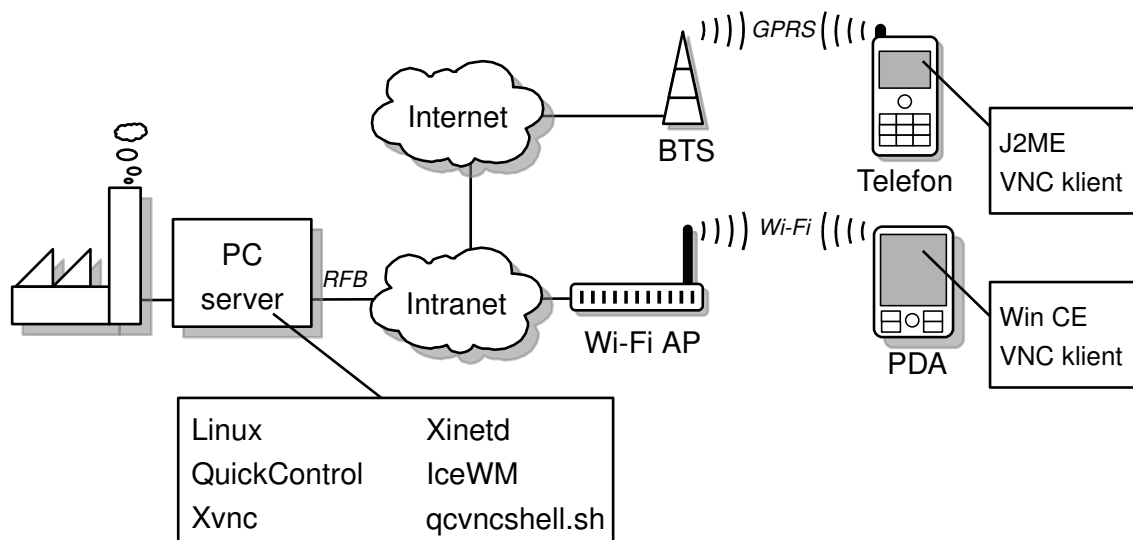
Kapitola 6

Návrh systému

Návrh systému vzdáleného řízení sestává ze dvou kroků. Nejprve je navržen způsob propojení jednotlivých částí systému popsaných v předchozí kapitole. Druhý krok spočívá v návrhu způsobu konfigurace systému pro konkrétní aplikaci. Vznikne tak zároveň referenční příručka pro člověka, který bude systém vzdáleného řízení nasazovat v reálném prostředí. Měla by ho krok za krokem provést konfigurací jednotlivých částí systému.

6.1 Schéma systému

Na obrázku 6.1 je vyobrazeno jedno z možných uspořádání systému pro měření a regulaci technologického procesu umožňujícího vzdálené řízení. Technologický proces je řízen počítačem, který kromě řídicího softwaru obsahuje také server Xvnc. Tak je umožněno řídit proces VNC klientům připojeným k lokální síti intranet. Je-li navíc lokální síť spojena se sítí internet, je možno proces ovládat například pomocí mobilního telefonu připojeného k internetu technologií GPRS.

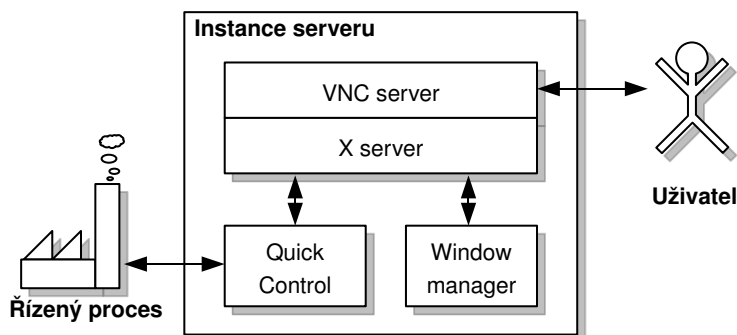


Obrázek 6.1: Celkové schéma systému vzdáleného řízení.

6.1.1 Instance serveru

Obrázek 6.2 ukazuje jednu instanci serveru. Instancí serveru se zde rozumí množina všech procesů, které se podílejí na realizaci jednoho spojení mezi klientem a serverem.

Faktické řízení technologického procesu je realizováno procesem QuickControl. Řízení probíhá na úrovni vnitřního protokolu QuickControl a není z hlediska vzdáleného řízení podstatné. QuickControl rovněž komunikuje prostřednictvím protokolu X s procesem Xvnc, který dále posílá po síti RFB protokolem obrazová data klientovi. Dále je zde proces IceWM, window manager poskytující komfortnější uživatelské rozhraní. Do zobrazení se zapojuje na úrovni X protokolu komunikací s Xvnc.



Obrázek 6.2: Schématické znázornění jedné instance serveru.

6.1.2 Xinetd

Jedna instance serveru umožňuje jednomu vzdálenému VNC klientovi řídit technologický proces. Obecně je ale potřeba mít možnost připojit klientů více současně. Z toho důvodu je pro realizaci serveru použit démon Xinetd popsáný v kapitole 5.2. Ten bude na určeném portu poslouchat místo serveru Xvnc. Dojde-li k požadavku na spojení, ověří oprávněnost požadavku a v případě úspěchu vytvoří novou instanci serveru. Dále pak poslouchá na zadaném portu. Viz obrázek 6.3.

Vzhledem k tomu, že démon Xinetd umožňuje při spojení spustit pouze jeden proces a architektura serveru vyžaduje běh tří procesů, bude nutno jejich spuštění zajistit jiným způsobem, například skriptem. V navrhovaném systému je tímto spouštěcím skriptem qcvncshell.sh, skript pro interpret Bash.¹

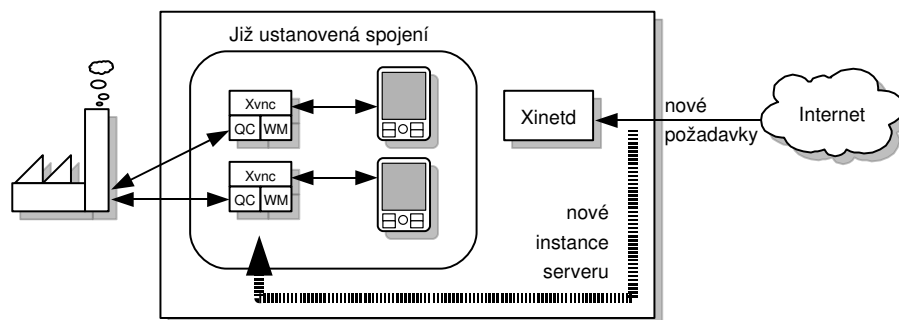
6.1.3 qcvncshell.sh

Skript qcvncshell.sh je spuštěn démonem Xinetd pro každou instanci serveru. Úkolem skriptu je:

1. Spustit Xvnc server.
2. Spustit QuickControl
3. Spustit IceWM

Podrobnosti implementace tohoto skriptu jsou uvedeny v kapitole 7.1.3.

¹Bash je unixový příkazový interpret naprogramovaný v roce 1987 v rámci projektu GNU.



Obrázek 6.3: Schéma ukazující, jak Xinetd vytváří nové instance serveru.

6.1.4 Časový průběh spojení

V předchozích odstavcích byl popsán princip realizace služby serveru vzdáleného řízení. Pro úplnost je zde uveden časový průběh spojení tak, jak se na něm podílejí výše zmíněné procesy.

1. Démon Xinetd na serveru poslouchá na určeném portu služby.
2. Klient se připojí na IP adresu serveru s číslem portu služby.
3. Xinetd kontroluje oprávněnost požadavku podle konfigurace služby. V případě, že je požadavek neoprávněný, spojení se ukončí.
4. Xinetd spouští skript `qcvncshell.sh`, na jeho standardní vstup a výstup připojí síťový soket spojení.
5. Xinetd dále poslouchá na portu služby a čeká na další požadavky.
6. `qcvncshell.sh` spouští server Xvnc.
7. `qcvncshell.sh` spouští QuickControl s parametrem displeje, na kterém se má vykreslovat.
8. `qcvncshell.sh` spouští window manager IceWM s parametrem displeje, na kterém se má vykreslovat.
9. `qcvncshell.sh` čeká na proces QuickControl. Po jeho skončení ukončí běh serveru Xvnc.
10. Při zrušení spojení mezi Xvnc a VNC klientem je Xvnc automaticky ukončen spolu se všemi jeho X klienty.

6.2 Způsob konfigurace

Konfigurace serverové části systému sestává z několika kroků, které budou popsány dále. Vzhledem k tomu, že různá přenosná zařízení mají různou velikost displeje, bude pravděpodobně nutné navrhnout každému typu zařízení zvláštní vizualizaci. Jen tak může být zajištěn optimální komfort v ovládní řízeného procesu. Musí také existovat mechanismus, kterým klientská aplikace serveru sdělí, kterou vizualizaci si přeje zobrazit. Nejsnazším způsobem, jak toho dosáhnout, aniž by bylo nutno měnit standardní komunikační protokol,

je nakonfigurovat Xinetd tak, aby přijímal požadavky na více portech a podle čísla portu spouštěl QuickControl s příslušnou vizualizací. Klient by tedy vhodnou vizualizací zvolil tím, že by se ke službě připojil na vhodném portu.

Každému typu zařízení s vlastní vizualizací tedy přísluší:

- 1 řádek v souboru `/etc/services`.
- 1 konfigurační soubor v adresáři `/etc/xinetd.d`.
- 1 spustitelný skript `qcvncshell.sh`
- 1 skript spouštějící QuickControl s příslušným modulem GUI (může být součástí `qcvncshell.sh`).

Na jednu stranu se může zdát konfigurace s nutností modifikace čtyř různých souborů pro každé zařízení příliš složitá a zdlouhavá. Na druhou stranu umožňuje ale takový přístup vysokou flexibilitu. Je například možno přidělit každému zařízení pevnou IP adresu a na jejím základě umožnit přístup k řízení technologie jen v určitou dobu, případně použitím speciální vizualizace umožnit řízení jen části technologického procesu. V takovém případě je navíc zaručena plná kontrola nad tím, z kterého zařízení – tedy kým – byl kdy proces řízen. Úpravy v jednotlivých konfiguračních souborech jsou většinou drobného charakteru. Výjimkou je konfigurace modulu GUI, kde je nutno brát v úvahu parametry přenosného zařízení a podle toho ručně upravit vizualizaci a ovládání. To je však nutné pouze pro zvýšení uživatelského komfortu. V případě potřeby je vždy možno navrhnout jednu univerzální vizualizaci pro všechna přenosná zařízení.

6.2.1 Instalace softwaru

Před konfigurací systému vzdáleného řízení je nutno se přesvědčit, zda jsou na serveru nainstalovány následující softwarové součásti:

- Unixový operační systém
- QuickControl s modulem GUI
- Xvnc server
- Démon Xinetd
- Window Manager (například IceWM)
- Interpret bash
- Nastavený uživatelský účet s příslušnými přístupovými právy
- `qcvncshell.sh` spustitelný zvoleným uživatelem

6.2.2 Konfigurace QuickControl

Ke konfiguraci systému QuickControl slouží nástroj QCeditor, který produkuje konfigurační soubory ve formátu XML. S jeho pomocí lze nakonfigurovat veškeré moduly, které systém QuickControl obsahuje a které jsou potřeba pro řízení daného technologického procesu. Z hlediska vzdáleného řízení je podstatná pouze konfigurace modulu GUI – modulu, který umožňuje vizualizaci dat. Konfigurace QuickControl z hlediska vzdáleného řízení sestává z následujících kroků:

1. Zjistit parametry zvoleného přenosného zařízení. Jedná se jednak o velikost displeje, jak fyzickou, tak v počtu obrazových bodů. Dále jsou důležité možnosti přístroje zpracovat události uživatelského vstupu.
2. Nakonfigurovat modul GUI podle vlastností zařízení. QCeditor umožňuje realizovat nejen libovolné rozmístění vizualizačních komponent a jejich přizpůsobení (velikost písma, barva, otočení), ale také přizpůsobení uživatelského ovládání. Pokud například není na přístroji přítomno polohovací zařízení, může být pro navigaci po jednotlivých ovládacích prvcích použito pouze kláves.
3. Vytvořit skript, který spustí QuickControl s požadovaným GUI modulem. Jedná se většinou o velmi jednoduchý skript sestávající z jednoho či dvou příkazů. V případě potřeby je možno celou jeho funkčnost přesunout do příslušného skriptu qcvncshell.sh

6.2.3 Konfigurace Xvnc

Xvnc je na serveru spouštěn skriptem qcvncshell.sh a všechny potřebné parametry budou předány jako argumenty na příkazové řádce. Jediné, co je potřeba nakonfigurovat je heslo VNC serveru. To je standardně hledáno v souboru `.vnc/passwd` v domácím adresáři uživatele.

Heslo lze vytvořit spuštěním programu `vncpasswd` s parametrem cesty k souboru, kam má být heslo uloženo. Není-li parametr zadán, uloží se heslo do souboru `~/.vnc/passwd`. Program `vncpasswd` vyžaduje zadat heslo alespoň 6 znaků dlouhé. Pokud `vncpasswd` není na serveru nainstalován, může být spuštěn na jiném počítači a soubor s heslem na server zkopírován.

6.2.4 Konfigurace Xinetd

Konfigurace služby Xinetd sestává z následujících kroků:

1. Přihlásit se jako superuživatel příkazem `su`.
2. Do souboru `/etc/services` doplnit řádek s číslem portu, na kterém bude Xinetd očekávat spojení. Tedy například:

```
xvnc 5950/tcp
```

Tento krok je možno vynechat, pokud budou v konfiguraci služby uvedeny následující parametry:

```
type = UNLISTED
port = číslo_portu
```

3. Do souboru `/etc/xinetd.conf` doplnit cestu k souboru, v němž je uložena konfigurace pro Xinetd. Například

```
include /home/asyc/xvnc
```

Případně je možno konfigurační soubor uložit do adresáře `/etc/xinetd.d`, odkud je automaticky načten v případě, že soubor `/etc/xinetd.conf` obsahuje direktivu

```
includedir /etc/xinetd.d
```

4. Upravit konfigurační soubor Xinetd (/etc/xinetd.d/) podle aktuálních požadavků. Pro základní funkčnost jsou důležité následující parametry:

```
service xvnc
{
    protocol      = tcp
    socket_type   = stream
    wait          = no
    user          = asyc
    server        = /home/asyc/qcvncshell.sh
}
```

Soupis dalších argumentů konfiguračního souboru je uveden v kapitole 5.2.3, kompletní seznam lze najít v manuálových stránkách (man xinetd.conf).

5. přinutit službu Xinetd k novému načtení konfigurace

```
service xinetd reload
```

6.2.5 Konfigurace window manageru

Konfigurace window manageru závisí na tom, jaký window manager je zvolen. V konfiguraci je obvykle možno zvolit různá nastavení dekorace oken, zobrazení/skrytí panelu aplikací, apod.) Cílem konfigurace pro použití na VNC je minimalizovat vykreslování dekorací na nezbytné minimum a maximalizovat zobrazovanou plochu (skrýt nabídky, panely, apod.). V případě použití window manageru IceWM je na výskyt konfiguračních souborů implicitně prohledáván adresář *.icewm* v domovském adresáři uživatele.

Window manager není nutnou součástí serveru. V případě jednoduché vizualizace, jejíž součástí nejsou žádná dialogová okna nemusí být window manager vůbec přítomen.

6.2.6 Konfigurace skriptu qcvncshell.sh

Konfigurace skriptu qcvncshell.sh spočívá v modifikaci skriptu samotného. V jeho úvodní části je definice několika proměnných, které jsou dále ve skriptu použity. Jejich změnou lze dosáhnout změny funkčnosti skriptu. Jsou to především:

<code>logfile</code>	Cesta k souboru pro zaznamenávání událostí. Implicitně qcvncshell.log.
<code>loglength</code>	Počet řádků souboru pro zaznamenávání událostí.
<code>xvnc</code>	Cesta ke spustitelnému souboru serveru Xvnc.
<code>xvncparams</code>	Parametry příkazové řádky pro server Xvnc.
<code>QC</code>	Cesta ke skriptu pouštějícího QuickControl.
<code>QCparams</code>	Parametry, které se mají předat skriptu \$QC.
<code>WM</code>	Cesta ke spustitelnému souboru window manageru.
<code>WMparams</code>	Parametry příkazové řádky pro window manager.

Kapitola 7

Implementace

Implementace systému vzdáleného řízení sestává z implementace serveru a implementace klienta. Všechny použité konfigurační soubory serveru a skript `qcvncshell.sh` jsou umístěny na příloženém CD v adresáři `server`. Zdrojové kódy klientských aplikací jsou na CD umístěny v adresáři `klient`.

7.1 Server

Funkčnost serverové části systému spočívá z velké části v konfiguraci jednotlivých komponent serveru, tedy QuickControl, Xinetd, Xvnc a IceWM. To je práce, která bude muset být zřejmě udělána s každým novým nasazením systému vzdáleného řízení, neboť je aplikačně specifická. Zde bude server nakonfigurován pro účely demonstrace jeho možností. Výsledkem bude plně funkční demo aplikace přístupná po nějakou dobu v síti internet na IP adrese `62.168.53.50:5950`.

Druhá část implementace serveru spočívá v implementaci skriptu `qcvncshell.sh`. Ten obstarává každou instanci serveru a je rovněž konfigurovatelný podle aktuálních potřeb.

7.1.1 Konfigurace Xinetd

Způsob konfigurace démona Xinetd je popsána v kapitole [6.2.4](#). Zde je již uveden pouze obsah jednotlivých konfiguračních souborů.

`/etc/services`

Služba je nakonfigurována na portu 5950. Toho je dosaženo následujícím řádkem v souboru `/etc/services`. V případě, že by tento krok konfigurace byl vynechán, bylo by nutno odkomentovat poslední dva řádky v konfiguračním souboru `/etc/xinetd.d/xvnc`. Viz dále.

```
xvnc          5950/tcp
```

`/etc/xinetd.d/xvnc`

Význam jednotlivých položek konfiguračního souboru uvedeného níže je následující: Zaznamenávání událostí je nastaveno do souboru `/home/asyc/qcvnc.log`, v případě úspěšného spojení klienta se serverem se do log-souboru zapíše IP adresa klienta, id procesu serveru a délka spojení. V případě neoprávněného požadavku je zaznamenána IP adresa, odkud bylo o službu neúspěšně žádáno. Jako server je spuštěn skript `/home/asyc/qcvncshell.sh`

pod uživatelským účtem asyc. Služba je dostupná denně od 6. hodiny ráno do 10. hodiny večerní. Maximální počet připojených klientů je 3, z jedné IP adresy však jenom jeden. Není dovoleno se připojit z IP adresy 217.115.241.21.

```
service xvnc
{
    log_type          = FILE /home/asyc/qcvnc.log
    log_on_success    += HOST PID DURATION
    log_on_failure    += HOST
    protocol          = tcp
    socket_type       = stream
    wait              = no
    user              = asyc
    server            = /home/asyc/qcvncshell.sh
    access_times      = 6:00-22:00
    instances         = 3
    per_source        = 1
    no_access         = 217.115.241.21
    # type            = UNLISTED
    # port            = 5950
}
```

7.1.2 Konfigurace IceWM

Konfigurace IceWM spočívá ve skrytí panelu pro spouštění aplikací:

```
ShowTaskBar=0
```

7.1.3 qcvncshell.sh

Celý skript qcvncshell.sh je k dispozici na CD, které je součástí této práce. Zde je jen stručně popsán algoritmus s ukázkou důležitých částí skriptu:

1. Je spuštěn server Xvnc. Po expanzi proměnných vypadá provedený příkaz asi takto:

```
/usr/X11R6/bin/Xvnc -desktop ASYC -nocursor -inetd -geometry 480x640
```
2. Příkazem `xvncpid=$!` je uschováno číslo procesu Xvnc serveru. Podle něj je pak nalezen displej, na němž Xvnc očekává obrazová data. (Prohledávají se soubory `/tmp/.X[0-9]+-lock`).
3. Příkazem `export DISPLAY=$N` je uložena systémová proměnná DISPLAY která slouží dále spouštěným programům k informaci o tom, ke kterému displeji se mají připojit a vykreslovat do něj.
4. Je spuštěn QuickControl: `/home/asyc/mp/QC/QCruntime/linux/asyc_client_111`.
5. Příkazem `qcpid=$!` je uschováno číslo procesu QuickControl.
6. Je spuštěn window manager IceWM: `/usr/bin/icwm`.
7. Čeká se na ukončení procesu QuickControl: `wait $qcpid`
8. Po skončení QuickControl je ukončena i činnost serveru Xvnc: `kill $xvncpid`

7.2 Klient

Klientská část systému pokrývá širokou škálu přenosných zařízení. Většina současných přenosných zařízení a operačních systémů podporuje technologii Java MIDletů. Výjimku tvoří operační systémy odvozené od platformy Windows, kde je běh Java aplikací buď problematický, nebo drahý. Proto je součástí implementace také klient v nativním kódu pro Windows Mobile.



Obrázek 7.1: Simulátor mobilního telefonu s přihlašovací obrazovkou VNC klienta.

7.2.1 Windows Mobile klient

Implementace klienta pro Windows Mobile vychází ze softwaru PocketPC VNCViewer Client, jehož autorem je Mark Midgley. Je šířený pod licencí GPL, což umožňuje modifikaci zdrojového kódu a následnou redistribuci pod licencí GPL. Internetové stránky produktu jsou na adrese <http://www.cs.utah.edu/~midgley/wince/vnc.html> Zdrojové kódy jsou dostupné na požádání autora.

Úpravy zdrojového kódu klienta pro Windows Mobile spočívaly nejen v úpravě vzhledu (doplnění loga produktu QuickControl, změna popisků, apod.), ale také v optimalizaci výkonu aplikace. Aplikace byla optimalizována pro minimální požadavky na velikost operační paměti přístroje. Veškeré vykreslování na obrazovku probíhalo použitím GDI¹ funkce Put-Pixel, která na obrazovku zařízení vykreslí jeden obrazový bod. Vykreslování bod po bodu je velmi neefektivní, proto byl algoritmus vykreslování přepsán tak, že byla nejprve v paměti vytvořena bitová mapa, která pak byla najednou vykreslena na obrazovku. Tím bylo dosaženo výrazného zvýšení výkonu aplikace za cenu zvýšeného nároku na množství

¹GDI je zkratkou pro Graphics Device Interface a je základním rozhraním používaným v operačních systémech Windows pro přístup ke grafickým objektům a jejich přenosu na výstupní zařízení.[8]

použité paměti. V použitém zařízení není zvýšení paměťového nároku aplikace z hlediska výkonu významné.

K úpravě zdrojových kódů a překladu aplikace do binární podoby spustitelné na zařízení s operačním systémem Windows Mobile byl použit nástroj Microsoft eMbedded Visual C++ 4.0, který společnost Microsoft nabízí volně ke stažení ze svých webových stránek. (<http://www.microsoft.com/downloads/>)

7.2.2 Java klient

Implementace Java klienta vychází ze softwaru J2ME VNC. Jeho autorem je Michael Lloyd Lee, použita je také licence GPL. Zdrojové kódy jsou dostupné z internetové stránky produktu – <http://j2mevnc.sourceforge.net/>.

Úpravy zdrojového kódu Java klienta spočívaly pouze v modifikaci vzhledu. K úpravě byl použit vývojový nástroj Sun Java Wireless Toolkit.²

Na obrázku 7.1 je snímek obrazovky počítače s emulátorem přenosného zařízení, na němž běží aplikace VNC klient.

²JWTK dostupný z webu společnosti SUN – <http://java.sun.com/products/sjwtoolkit/>

Kapitola 8

Závěr

V sídle firmy ASYC bylo odzkoušeno zapojení podle schématu 6.1 z kapitoly 6.1. Byla ověřena funkčnost Wi-Fi spojení serveru s klientem pro Windows Mobile běžícím na přístroji Acer n300. Server byl rovněž připojen k síti internet a bylo otestováno připojení přes GPRS za použití několika výkonnostně odlišných zařízení.

Řešení bylo shledáno plně funkčním za použití PDA a spojení Wi-Fi. Při připojení pomocí mobilního telefonu je komfort použití vzdáleného řízení menší. Je to způsobeno nižším výkonem zařízení a pomalým připojením GPRS (dlouhá doba odezvy), menšími rozměry displeje (nižší přehlednost) a absencí polohovacího zařízení (obtížná dostupnost ovládacích prvků). Do budoucna lze však očekávat větší dostupnost technologií, které jsou vyvinuty již dnes, jsou však pro běžné použití ještě příliš drahé. To pravděpodobně povede ke zvýšení použitelnosti systému vzdáleného řízení spolu s běžně dostupnými mobilními telefony.

Budoucí vývoj systému bude zřejmě spočívat ve sledování vývoje situace na trhu mobilních zařízení a údržby funkcionality produktu ve spojení s novými dominantními hardwarovými a softwarovými technologiemi.

Literatura

- [1] Wikipedia Community. Adobe flash. http://en.wikipedia.org/wiki/Adobe_Flash 8. 1. 2008.
- [2] Wikipedia Community. Ajax (programming). <http://en.wikipedia.org/wiki/AJAX> 19. 5. 2008.
- [3] Wikipedia Community. Arm architecture. http://en.wikipedia.org/wiki/ARM_architecture 8. 1. 2008.
- [4] Wikipedia Community. Blackberry. <http://en.wikipedia.org/wiki/Blackberry> 8. 1. 2008.
- [5] Wikipedia Community. Dynamic html. <http://en.wikipedia.org/wiki/DHTML> 8. 1. 2008.
- [6] Wikipedia Community. Enhanced data rates for gsm evolution. http://en.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution 8. 1. 2008.
- [7] Wikipedia Community. General packet radio service. <http://en.wikipedia.org/wiki/GPRS> 8. 1. 2008.
- [8] Wikipedia Community. Graphics device interface. http://en.wikipedia.org/wiki/Graphics_Device_Interface 19. 5. 2008.
- [9] Wikipedia Community. High-speed circuit-switched data. http://en.wikipedia.org/wiki/High-Speed_Circuit-Switched_Data 8. 1. 2008.
- [10] Wikipedia Community. High-speed downlink packet access. <http://en.wikipedia.org/wiki/HSDPA> 8. 1. 2008.
- [11] Wikipedia Community. Html. <http://en.wikipedia.org/wiki/HTML> 8. 1. 2008.
- [12] Wikipedia Community. Hypertext transfer protocol. <http://en.wikipedia.org/wiki/HTTP> 8. 1. 2008.
- [13] Wikipedia Community. Java platform, micro edition. <http://en.wikipedia.org/wiki/J2ME> 8. 1. 2008.
- [14] Wikipedia Community. Mac os x. http://en.wikipedia.org/wiki/Mac_os_x 8. 1. 2008.
- [15] Wikipedia Community. .net compact framework. http://en.wikipedia.org/wiki/.NET_Compact_Framework 8. 1. 2008.

- [16] Wikipedia Community. Palm os. http://en.wikipedia.org/wiki/Palm_os 8.1.2008.
- [17] Wikipedia Community. Personal digital assistant. http://en.wikipedia.org/wiki/Personal_digital_assistant 8.1.2008.
- [18] Wikipedia Community. Programmable logic controller. http://en.wikipedia.org/wiki/Programmable_logic_controller 8.1.2008.
- [19] Wikipedia Community. Rdp. http://en.wikipedia.org/wiki/Remote_Desktop_Protocol 19.5.2008.
- [20] Wikipedia Community. Smartphone. <http://en.wikipedia.org/wiki/Smartphone> 8.1.2008.
- [21] Wikipedia Community. Symbian os. <http://en.wikipedia.org/wiki/Symbian> 8.1.2008.
- [22] Wikipedia Community. Virtual network computing. <http://en.wikipedia.org/wiki/VNC> 8.1.2008.
- [23] Wikipedia Community. Wifi. <http://en.wikipedia.org/wiki/WiFi> 8.1.2008.
- [24] Wikipedia Community. Window manager. http://en.wikipedia.org/wiki/Window_manager 19.5.2008.
- [25] Wikipedia Community. Windows mobile. http://en.wikipedia.org/wiki/Windows_mobile 8.1.2008.
- [26] Wikipedia Community. X window system. http://en.wikipedia.org/wiki/X_window_system 19.5.2008.
- [27] Wikipedia Community. X window system protocols and architecture. http://en.wikipedia.org/wiki/X_Window_System_protocols_and_architecture 19.5.2008.
- [28] David Häring. Xinetd – vysoce konfigurovatelný super-server. <http://www.linuxzone.cz/index.phtml?ids=29&idc=134> 19.5.2008.
- [29] Petr Lesný. *Úvod do programování PalmOS*. Vlastním nákladem, 2003.
- [30] Tristan Richardson RealVNC Ltd. The rfb protocol. <http://www.realvnc.com/docs/rfbproto.pdf> 8.1.2008.
- [31] Kim Topley. *J2ME v kostce*. Grada publishing, 2004. ISBN 80-247-0426-9.
- [32] Komunita Wikipedie. Bluetooth. <http://cs.wikipedia.org/wiki/Bluetooth> 8.1.2008.
- [33] Ľuboslav Lacko. *Vývoj aplikácií pre mobilné zariadenia*. Microsoft, 2002.
- [34] Ľuboslav Lacko. *Programujeme mobilní aplikace ve Visual Studiu .NET*. Computer Press, 2004. ISBN 80-251-0176-2.