

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Systém pro usnadnění správy SW a řízený výběr aplikací

Plzeň 2007

Mgr. Filip Vaculík

Zadání práce

Poděkování

Dnešní svět, snad více než kdy dřív, podporuje individualismus. Může to svádět k myšlence, že s mobilem a připojením k Internetu lze zvládnout všechno. Že vlastně nikoho nepotřebujeme. Ale je to opravdu jen zdání.

Rád bych poděkoval lidem, kteří tu kolem mě žijí, za to, že jsou rádi když se mi daří a za jejich pomoc, když se zrovna nedaří. Bez toho by to celé snažení, nedávalo smysl.

Poděkování si zaslouží i lidé z fakulty, obzvláště ti, kteří i v dnešní uspěchané době věnují svůj čas nám, studentům a snaží se nás dostat výš.

Poděkování posílám i té obrovské skupině vývojářů open source, bez jejich práce si tu svou nedovedu ani představit.

Abstrakt

Methods for simplification software administration and managed applications selection.

Theoretical part of diploma thesis inquires into characteristic operations in software administration. The result is presented as a report about helpful procedures and serviceable utilities with links to it's sources. Focus is also given on complex software products, which covered whole or main part of software deployment live cycle. Note is given to standardization activities known as companies and groups. The range of research includes both, proprietary and open source realms. Part of the thesis engages in design systems for administration software and development multiplatform as well as modular system core, with add-on for sorting links and its execution

Obsah

1 Úvod	8
1.1 Proč jsem si zvolil toto téma	8
1.2 Členění diplomové práce	8
2 Problematika správy SW	9
2.1 Z historie	9
2.2 Pohledem zúčastněných	10
2.3 Charakteristické činnosti v systému správy SW	10
2.3.1 Nalezení vhodného SW	11
2.3.2 Získání SW	13
2.3.3 Skladování a evidence	15
2.3.4 Příprava instalace a repackaging	17
2.3.5 Instalace	20
2.3.6 Konfigurace	22
2.3.7 Zpřístupnění a znepřístupnění aplikace	24
2.3.8 Výběr a spouštění	24
2.3.9 Používání aplikací	24
2.3.10 Opravy SW, reinstalace	25
2.3.11 Migrace aplikace	26
2.3.12 Upgrade, aktualizace a doplňky	27
2.3.13 Odinstalace	28
3 Existující komplexní řešení správy SW	29
3.1 V prostředí OS MS Windows	29
3.1.1 Microsoft – Systems Management Server	29
3.1.2 Landesk	30
3.1.3 Další řešení systému správy SW	31
3.2 V prostředí OS Linux	31
3.2.1 Založené na RPM	32
3.2.2 Založené na DEB	33
3.2.3 Specializované	33
3.3 V prostředí OS BSD	35
3.4 V prostředí OS Solaris	36
3.5 V prostředí MAC OS	36
3.6 Multiplatformní	37
3.6.1 Novell – ZENworks	37
3.6.2 Postavené na jazyce Java	37
3.6.3 Postavené na jazyce Python	38
3.6.4 Postavené na jazyce C nebo C++	38
3.6.5 Virtuální prostředí	38
3.7 Zhodnocení s ohledem na možnost využití v prostředí základních a středních škol	39
3.8 Standardizační snahy v oblasti systému správy SW	40
3.8.1 Linux Standard Base	40

3.8.2 Solution Deployment Descriptor	41
3.8.3 Software–Based Communication – Domain Task Force	41
3.8.4 JSR 88: Java™ EE Application Deployment	41
4 Návrh systému pro usnadnění správy SW	42
4.1 Výchozí předpoklad	42
4.2 Zjištěná situace	42
4.3 Návrh obecných vlastností	42
4.4 Úroveň jednotlivých stanic	43
4.4.1 Úroveň lokální sítě	45
4.4.2 Úroveň organizace	47
4.4.3 Úroveň pro výměnu dat mezi organizacemi	48
4.5 Modul pro usnadnění spouštění aplikací	48
4.5.1 V prostředí Linuxu	49
4.5.2 V prostředí MS Windows	50
4.6 Závěr návrhu systému pro správu SW	51
5 Implementace systému pro správu SW	52
5.1 Výběr programovacího jazyka a vývojových nástrojů	52
5.2 Aplikace poznatků SW inženýrství při vývoji	53
5.2.1 Model–view–controller architektura	53
5.2.2 Dokumentace	53
5.2.3 Automatické testy	53
5.2.4 Internacionalizace a lokalizace	54
5.2.5 Logování	54
5.2.6 Správa verzí	54
5.2.7 Refactoring	54
5.3 Jádro systému	55
5.3.1 Možnosti detekce HW a prostředí OS	56
5.3.2 Možnosti úprav nastavení prostředí	57
5.4 Rozšíření – vyhledání zástupců aplikací	57
5.5 Rozšíření pro vzdálené spouštění aplikací	57
5.6 Rozšíření – GUI pro třídění a spouštění aplikací	58
5.7 Nasazení v praxi	58
6 Závěr	59
Přehled zkratk	60
Slovník	61
Přílohy	64

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím uvedených pramenů.

V Plzni dne 15. května 2007

Filip Vaculík

1 Úvod

Správa software je běžnou činností, se kterou má zkušenost téměř každý uživatel PC. Je nezbytně nutná, pokud chceme rozšiřovat možnosti, které nám naše počítače poskytují. Neobejdeme se bez ní, pokud chceme v našem počítači udržovat funkční prostředí. Stále se zvyšuje počet stanic v komerčních firmách, státních organizacích i domácnostech. Trvale přibývá počet nově vytvářených a dostupných aplikací. Přibývá propojení výpočetní techniky s činnostmi běžného života. Fungování naší společnosti je na prostředcích informačních technologií stále závislejší.

Jakkoliv je tato oblast významná, obecné povědomí o správě software je na velmi nízké úrovni. V celé vzdělávací soustavě České republiky není této oblasti věnována dostatečná pozornost. Běžně je správa software postavena výhradně na základních nástrojích, dodávaných spolu s operačním systémem, které jsou na hraně svých možností. Domnívám se, že jde o oblast kritickou, která vyžaduje výzkum a aplikaci nových koncepcí. Na rozvoji správy software závisí rozvoj mnoha dalších oblastí. Svou prací bych rád přispěl ke zvyšování povědomí.

1.1 Proč jsem si zvolil toto téma

Od roku 2002 jsem hlavním systémovým administrátorem v několika státních a komerčních organizacích. Velká část povinností spočívá v přípravě a údržbě prostředí pro stovky uživatelů. Větší část připadá na správu SW. Věnoval jsem se i školitelské činnosti v této oblasti. Svou prací bych rád spojil pohled uživatele, administrátora a programátora na problematiku správy software. Očekával jsem, že systematictější průzkum této oblasti přinese obohacení o užitečné informace pro mě a snad i pro čtenáře této práce.

1.2 Členění diplomové práce

První část práce je pojata jako přehled postupů a doporučení vztahujících se k charakteristickým oblastem správy SW. Kapitoly jsou doplněny informacemi z praxe a odkazy na osvědčené programy. Druhá část je věnována představení komplexních řešení systémů správy SW a vývojových směrů v této oblasti. Třetí část se věnuje implementaci prototypu aplikace, která by se mohla stát základem modulárního systému správy SW. Fungování aplikace je prezentováno na modulu pro třídění zástupců a řízené spouštění aplikací.

2 Problematika správy SW

2.1 Z historie

V počátcích výpočetní techniky byly programy tak úzce svázány s konkrétním hardware, že správa SW se odehrávala spíše v rovině mechanické. V závislosti na paměťových médiích se spíše řešilo, kde skladovat, jak popisovat a jak organizovat přístup k strojovému času. Samotný běžící program pak měl k dispozici veškerý dostupný HW. Programátor byl často zároveň i uživatel. Administrátor se věnoval spíše fungování HW [1].

Již velmi brzy se však začaly k počítačům dodávat jednoduché knihovny základních funkcí, které se staticky linkovaly k programům. Spolu se zdokonalováním záznamových médií byly fronty lidí čekajících na přístup k počítačům nahrazeny scripty, které dávkově vykonávaly spouštění programů.

Dalším vývojovým krokem byly provozní (runtime) knihovny, které se zaváděly po startu počítače a obsahovaly funkce spouštění programů a úklid po skončení programu. Později, ve spolupráci s běžícím programem, dokázaly přepínat mezi více úlohami. Takovým programům se říkalo programové monitory.

Větší vlna zájmu o výpočetní techniku na konci 50. let, růst konkurence a posun v potřebách uživatelů motivovali výrobce dodávat k počítačům stále lepší základní programy. Zde lze najít zárodky operačních systémů, které však byly, především díky HW různorodosti, vytvářeny znovu pro každou modelovou řadu.

Výrazná změna se udála v roce 1964, kdy firma IBM oznámila koncepci série počítačů System/360 s operačním systémem OS/360 [2]. Počítače kompatibilní s touto sérií později vyrábělo i několik dalších firem v USA, Japonsku a dokonce i v Sovětském svazu. Hlavní změnou byla snaha o zachování zpětné kompatibility a možnost připravovat programy, které fungovaly na velkém množství počítačů. Tím byl otevřen prostor pro firmy, které se mohly specializovat na vývoj a prodej SW.

Oddělení jádra operačního systému od kódu ovladačů a uživatelského prostoru spolu s rozvojem překladačů a vyšších programovacích jazyků – to vše pomáhalo vytvořit vhodné prostředí a začalo nebývale růst množství uživatelských aplikací. Na začátku 80. let byl již trend zachovávání zpětné kompatibility dobře upevněn a výrazně urychlil rozšíření osobních počítačů. Programy se začaly objevovat v regálech supermarketů.

V současnosti není neobvyklé, že mezi aplikací a HW jsou kromě operačního systému i další vrstvy, jako například emulátory, virtuální prostředí a stroje. Některé programy lze v nezměněné podobě spouštět i po desítkách let. Existují tisíce firem, tisíce skupin i jednotlivců, produkujících nové aplikace. Rozvoj Internetu přináší nové možnosti i potřeby.

2.2 Pohledem zúčastněných

Na začátku by bylo vhodné, podívat se alespoň zběžně, jaké jsou potřeby a přání těch, kteří přímo s programy pracují. Nebudeme sledovat finanční stránku a bude to velmi idealistický pohled.

Programátor

Rád by jednoduše vytvořil program, který bude mít požadované funkce. Chtěl by, aby takový program mohl snadno dále vylepšovat, aby fungoval rychle, bezchybně, nevyčerpával příliš zdroje počítače, aby byl vizuálně pěkný, snadno se ovládal, aby mohl fungovat na všech platformách, ve všech operačních systémech, a aby splňoval všechny potřeby uživatelů.

Administrátor

Nejlepší hasič je ten, který provede taková opatření, aby jen obtížně došlo k požáru. Nejlepší administrátor je ten, který provede taková opatření, že vše funguje ke spokojenosti uživatelů, a neřeší žádné kritické situace. Měl by mít nástroje, kterými by snadno připravil systém a základní prostředí. Programy pro celou organizaci by instaloval na pár příkazů či kliknutí. Nemusel by řešit žádné záludné a neprůhledné problémy. Zdokonaloval by svou odbornou úroveň, aby mohl správně řešit nestandardní situace a požadavky na další rozvoj.

Uživatel

Dalo by se říci, že toho nechce mnoho. Chce mít dostupné programy a dokumenty se kterými pracuje. V podstatě mu vůbec nezáleží na tom, co má za počítač, jaký má operační systém a jak to všechno funguje. Chce používat funkce a postupy které zná. V případě že bude potřebovat nějaké další funkce, aby je mohl využívat, s minimální námahou se něco učít. Nemá zájem řešit fungování, nebo bezpečnost systému. Chce ho jen používat.

2.3 Charakteristické činnosti v systému správy SW

Na skutečném začátku vzniku aplikací najdeme potřeby. Potřebu usnadnit nebo umožnit nějakou činnost, potřebu vydělat peníze, potřebu prosadit se v komunitě a spoustu dalších potřeb. Tam, kde tyto potřeby najdou živnou půdu, následují rozbor, analýzy, specifikace, plánování, vyjednávání. Pokud aplikace projde touto fází, čeká na ni plejáda vývojových cyklů, alfa verze, beta verze a za nimi snad dojde i k samotnému vydání (release). Celá zmíněná část může být velmi jednoduchá a může proběhnout během pár hodin. Může však také trvat roky, případně v kterékoliv fázi neslavně skončit.

Z pohledu administrátora, pokud se zrovna nejedná o software na zakázku firmy, jde spíše o „prenatální stádium“, do kterého nemůže mnoho zasahovat. Je možné se zapojit do beta testů, případně cvičně nasadit prerelease, ale týká se to jen omezeného množství aplikací. Skutečný život aplikace začíná pro administrátora požadavkem konkrétní aplikaci zprovoznit, nebo zajistit určité služby.

Následující podkapitoly se věnují charakteristickým činnostem v životním cyklu nasazení (deployment) aplikací. Každé z těchto období přináší některé obtíže a rizika. Upřednostňovány budou finančně nenáročné programy a postupy.

2.3.1 Nalezení vhodného SW

U velkých SW produktů, za kterými stojí silná firma, bývá někdy téměř nemožné jejich aplikace či nové verze přehlédnout. Existenci byť velmi kvalitního, ale úzce specializovaného SW produkovaného jednotlivcem, případně malou skupinou, naopak bývá velmi obtížné zachytit. Mezi těmito extrémy je velká řada nuancí. Můžete zastávat názor, že administrátor se nemusí o tohle zajímat, že vše, o co se máte starat, už máte. Má zkušenost je však taková. Právě hledáním a testováním nových programů a orientací v široké nabídce lze ušetřit statisícové částky na pořizovacích cenách. Stejně tak lze i výrazně snížit celkové náklady na vlastnictví (TCO). V některých případech jde o jedinou možnost, jak získat nástroj k provádění specializované činnosti.

Vydáte-li se na cestu hledání vhodného SW, budete konfrontováni s velmi rozsáhlou oblastí. Naštěstí v tom nejste sami a zdatným pomocníkem vám mohou být odborné časopisy, servery, tematické diskuze. Odkazy na tematické zdroje viz Tabulka 4.

Vydání aplikace nebo nové verze

O některých aplikacích víte již pár let před jejich vydáním, k většině se dostanete až po jejich vydání. Doporučuji nehnat se přehnaně za nejnovějšími verzemi. Stabilní a známé prostředí je pro uživatele mnohem víc, než pár nových funkcí, které možná ani nevyužijí. Obzvláště je to rizikové u klíčových, často používaných programů. Časté inovace lze provozovat u programů, které používáte jen vy, nebo malá skupina uživatelů, a nemají velký vliv na chod systému. Jiná situace je v případě bezpečnostních oprav, i když i jejich aplikace přináší rizika, většinou je větším rizikem je neaplikovat.

Pokud si můžete určit načasování nasazení nových programů, vhodná doba bývá v průběhu větších podnikových dovolených. Uživatelé bývají odpočatí a přístupnější změnám. Opakující se rady lze shrnout do seznamu častých dotazů (FAQ) a to, pokud se stane používaným nástrojem, může ušetřit hodně práce.

Vhodné je se orientovat v problematice označování verzí programů, umožní vám to lépe načasovat případný přechod. Základní pojmy z oblasti verzování viz Tabulka 1.

pre-alpha, alpha, night build,, working copy, experimental branch, development release	Označení pro verze určené téměř výhradně vývojářům a pro testování
beta , preview, prototype, technical preview, early access	Většinou již program obsahuje všechny funkce a přechází do závěrečného testování
pre-release, release candidate, gamma, delta	Aplikace by neměla mít závažné chyby
stable, release, gold, final version	Hotová aplikace (nikoliv nutně bezchybná)

Tabulka 1: slovní označování verzí

Některé větší SW celky mají kromě verze své kódové jméno, např: MS Windows Vista, Gentoo Linux – Secret Sauce.

Velmi podstatný může být poznatek, že někdy vývoj předchozí větve SW nekončí vydáním nové verze, ale pokračuje. Například po verzi 2.6.2 grafické knihovny GTK+ přišly verze 2.7.x, 2.8.x, 2.9.x, ale zároveň byly vydávány verze 2.6.3, 2.6.4, a tak nyní (únor 2007) jsou aktuální stabilní verze GTK+ 2.9.4 a zároveň i 2.8.20 a 2.6.10. Na první pohled se může zdát, že jde o chybu, ve skutečnosti je to rozumný a chvályhodný postup, protože nové verze s sebou často přinášejí novou funkcionalitu, která se občas neobejde bez porušení zpětné kompatibility. Obzvláště je to důležité u SW, na kterém závisí funkcionalita jiných aplikací. V případě GTK+ to byl ve verzi 2.8 přechod na grafickou knihovnu Cairo, která nefunguje na MS Windows 98. Všichni s tímto OS, kteří by aktualizovali na tuto novou verzi, by způsobili nefunkčnost všech aplikací postavených na GTK+. Na druhou stranu, nemoci aktualizovat by znamenalo zřít se mnohých vylepšení a oprav, které nové verze GTK přinesly. Řešením tedy bylo udržovat více stabilních verzí najednou.

Případů nejasného verzování je mnoho, svého času zamotal lidem hlavu třeba i SUN s pojmenováním verzí Javy. Někdy to opravdu může být matoucí, ale bývá dobré se v tom zorientovat. Nejčastějším typem verzování bývá označování číselnou kombinací, oddělenou tečkami dle konvence: hlavní verze, podverze[, revize[, sestavení]] viz [3].

U nového programu vás pravděpodobně budou zajímat licence, tedy právní podmínky za jakých můžete SW používat, vaše práva a povinnosti z nabytí vyplývající, jakož i práva a povinnosti výrobce či distributora. Situace v této oblasti je dosti roztržitá a nepřehledná. Podle některých výkladů, obsahuje mnoho licencí tak závažné nedostatky, proti právním normám ČR, že jsou některá jejich ustanovení prakticky neplatná. Bohužel se to týká i licencí, pod kterými je šířen SW největších světových výrobců. Několik odkazů na výklad právních norem v této oblasti viz Tabulka 2.

Zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským < http://business.center.cz/business/pravo/zakony/autorsky/ >
Mgr. Petr Otevřel, <i>Dopady novely autorského zákona na oblast ICT</i> vydáno (25. 09. 2006) < http://www.pravoit.cz/view.php?nazevclanku=dopady-novely-autorskeho-zakona-na-oblast-ict&cislocclanku=2006090001 >
Dle právníků fy. Microsoft, mimochodem dosud (leden 2007) stále podle nenovelizovaného zákona z roku 2000 < http://www.microsoft.com/cze/pirastvi/pravo/autorskyzakon.asp > < http://www.microsoft.com/cze/pirastvi/legalnisw/licencni_smlouva.asp >

Tabulka 2: výklady k autorskému zákonu

O poznání lepší je pohled na typy licencí viz Tabulka 3.

http://www.slunecnice.cz/licence/	Stručný přehled některých SW licencí
http://www.gnu.cz/	Stránky věnující se jedné z nejrozšířenějších SW licencí GNU
<i>Licenční smlouva s koncovým uživatelem (EULA)</i> (leden 2007) < http://www.microsoft.com/cze/licence/ZakladniInformace/eula.msp >	
OEM licence – <i>Original equipment manufacturer</i> (leden 2007) < http://en.wikipedia.org/wiki/Original_equipment_manufacturer >	

Tabulka 3: softwarové licence

Je dosti podstatné, orientovat v tom, co to je: shareware, trial, addware, registerware, OEM. Jaký je rozdíl v pojmech freeware a open source a jaká se k jejich používání vážou pravidla.

Mnoho výrobců poskytuje výrazné slevy. Ty bývají vázány například na typ organizace, na množství odebíraných titulů či počet licencí. Mnoho velmi kvalitních aplikací je k dispozici zdarma. A pokud se vám podaří přesvědčit vaši organizaci, aby pár procent z ušetřené částky darovala na podporu open source, můžete pomoci jeho kvalitu dále zvyšovat.

2.3.2 Získání SW

Tradiční model distribuce SW pomocí externích paměťových médií hraje v oblasti komerčního SW stále prim. Je to podporováno i účetnictvím firem, kterému, zdá se, dělá obtíže, když nemůže pořízené zboží označkovat a do katalogu uvést, ve které místnosti se nachází. V dnešním zasítovaném světě jde často jen o mezikrok navíc. Mnohý SW, a téměř veškerý open source SW je distribuován prostřednictvím Internetu.

Velmi efektivně lze využívat služeb serverů věnujících se katalogizaci SW viz Tabulka 4.

http://www.slunecnice.cz http://www.studna.cz http://www.stahuj.cz http://www.gigamania.cz	České servery zaměřující se na obecnou katalogizaci SW
http://www.tucows.com/ http://www.download.com/	Zahraniční
http://www.apple.com/software/	SW pro počítače fy. Apple
http://www.linuxsoft.cz/ http://distrowatch.com	Specializující se na open source SW
http://mac.the-underdogs.info/ http://www.abandonia.com/	Orientující se na starý a opuštěný (abandonware) SW
http://www.driverzone.com/ http://www.driverguide.com/	Ovladač k HW
http://www.superwarehouse.com/software.cfm	Se zaměřením na komerční software produkty, rozříděné dle firem

Tabulka 4: servery zaměřující se na katalogizaci SW

Uvedený seznam je jen zlomkem skutečného množství serverů zabývajících se tímto tématem. Lze je využívat jako sklad programů, často mají možnost fulltextového vyhledávání a hierarchický katalog. SW zde uváděný bývá většinou prověřený a otestovaný, ale nelze na to spoléhat. Dobrým vodítkem může být hodnocení programu, recenze, ale užitečné mohou být i komentáře uživatelů. Tyto servery bývají dobře indexované vyhledávači a komentáře lze využít i k doplnění vlastních poznámek, případně v nich požádat o radu. Kromě toho, že tak můžete pomoci ostatním, mohou být vaše komentáře užitečné i vám.

Důležité údaje, které tyto servery o SW podávají, bývají: HW náročnost, OS, výrobce, licence, velikost, celkový počet stažení, počet stažení za poslední období, verze, datum poslední aktualizace, odkazy na alternativní programy, kategorie, podporované jazykové verze, odkaz na stažení. Nevýhodou bývá nedostupnost starších verzí programu a v případě, že samotný program není zrcadlen (mirroring) na úložišti serveru, jeho nedostupnost, pokud odkaz přestane platit. Je zde i mírné riziko podvrhu, pokud útočník pozmění na serveru výrobce distribuční soubory, nebo indexační soubor, ze kterého katalogy čerpají informace.

Ne vždy si však producent SW může dovolit disponovat dostatečně výkonnou linkou a některý SW je šířen pomocí sdílení dat v komunitních sítích. Nejčastěji se jedná o sítě založené na protokolu BitTorrent a klientských aplikací, které tento protokol používají. Základní myšlenkou fungování je, že zdrojová data jsou po menších kouscích rozeslána mezi stahující, kteří si tyto kousky vyměňují mezi sebou. Není tak nutné ze zdrojového serveru přenést veškerá data všem stahujícím, ale jen tolik, aby si je mohli aktuálně stahující mezi sebou vyměnit. Užitečné odkazy k tomuto tématu viz Tabulka 5.

Přehled klientských programů a jejich vlastností < http://en.wikipedia.org/wiki/Comparison_of_BitTorrent_software >	
Animace znázorňující princip fungování < http://en.wikipedia.org/wiki/Image:Torrentcomp_small.gif >	
http://www.bittorrent.com/	Oficiální stránka

Tabulka 5: odkazy k tematice BitTorrentu

Existuje však i mnoho dalších modelů a mechanismů distribuce SW. Výrazně lepší situace bývá mimo prostředí MS Windows. V mnohých distribucích operačních systémů postavených na Linuxu, BSD, Solarisu jsou k dispozici nástroje pro výběr, stažení a instalaci velkého množství aplikací. Více v kapitole o komplexních řešeních.

2.3.3 Skladování a evidence

Osobně se mi osvědčilo založit si archiv, ve kterém je SW roztržěn podle hlavních charakteristik. Po 10 letech používání má velikost 18GB, obsahuje téměř 5000 aplikací a velké množství dalších dat. 18GB se může zdát jako hodně, na celém serveru www.slunecnice.cz je v této době (leden 2007) 44GB dat. Ve skutečnosti by těch dat bylo mnohem více. Odhadem 400GB. CD s velkými aplikacemi zde uloženy nejsou. Jejich obsah je zde pouze indexován pomocí textových souborů ve kterých je .výčet obsahu CD.

Na rozdíl od katalogizačních serverů můžete ve vlastním archivu uchovávat více verzí téže aplikace a nejste odkázáni jen na jednu poslední. Nejnovější verze například nemusí podporovat prostředí, ve kterém ji chcete používat. Každá aplikace má samostatný adresář, ve kterém je i textový soubor s krátkým popisem aplikace, popisky s poznámkami. Je velmi cenné, když nemusíte opakovaně zkoumat nejasné chování či problémy aplikace, protože budete mít předchozí řešení již napsané. Pokud je v jedné kategorii aplikací více, bývá dobré do textových souborů uvádět i krátký identifikátor s hodnocením, případně dalšími charakteristikami. Pokud by některá aplikace logicky patřila na více míst v hierarchické struktuře, je umístěna pouze na jednom místě a na ostatních se uvede odkaz vedoucí k jejímu umístění. Někde je uložen i screenshot pro rychlé připomenutí toho, jak aplikace vypadá.

V podadresářích aplikace může být uložena i dokumentace, konfigurační skripty, log soubory, výpisy z registrů. Nejde o nic převratného, adresáře byly přeci právě pro tento účel vytvořeny, ale výhodou je, že pro práci s touto masou dat můžete použít konvenční, propracované a osvědčené nástroje. Základním nástrojem je vhodný správce souborů. Špičkou ve svém oboru je souborový manager Total Commander. Na Linuxu v textové konzoli Midnight commander, v GUI Krusader. Total commander je sice komerční, cena jedné licence je 28€, primárně je určen jen pro platformu MS Windows, ale stejně kvalitní nástroj jsem bohužel nenašel. Pro Total Commander je vyrobeno velké množství pluginů,

kteře dále rozšiřují jeho funkčnost. Na Linuxu může fungovat v emulátoru Wine, ale není připraven pracovat s právy souborů, ani s linuxovými linky. Odkazy k souborovým managerům, viz Tabulka 6.

Test osmi souborových managerů pro Linux < http://www.root.cz/clanky/osm-souborovych-manazeru/ >	
Kategorie správců souborů na Slunečnici, za povšimnutí stojí Servant salamander, DOS navigátor pro případ práce v DOSu, případně FAR, < http://www.slunecnice.cz/utility/usporadani-souboru/souborove-managery/ >	
http://mujweb.cz/www/cablik/diskman1.htm	Povedený souborový manager, vyrobený v Čechách, pro nekomerční využití zdarma
http://ghisler.com/	Domovská stránka Total Commanderu
http://ghisler.com/addons.htm	Stránka s pluginy pro Total Commander
Popis některých funkcí Total Commanderu < http://www.emag.cz/pod-poklickou-total-commanderu/ >	
http://totalcmd.pl/	Total Commander s předpřipravenými pluginy
Popis programu Midnight Commander < http://www.strojsnv.sk/linux/programy/mc.htm >	

Tabulka 6: managery souborů

V následující tabulce uvádím několik odkazů na aplikace pro práci s archivy dat.

http://sourceforge.net/projects/sevenzzip/	Multiplatformní kompresní program
http://www.snapfiles.com/get/cmdmirror.html	Program pro synchronizaci adresářů z příkazové řádky pod MS Windows
http://samba.anu.edu.au/rsync/	Synchronizační program pro Linux s obrovským výběrem možností
Seriál o komprimování v Linuxu < http://www.root.cz/serialy/unixova-komprese-v-praxi/ >	

Tabulka 7: utility pro správu souborů

Je užitečné mít k vlastnímu archivu programů přístup z jakéhokoliv počítače v síti. Pro přístup ze vzdálenějších míst například pomocí FTPS serveru, pro použití v lokální síti je pohodlnější sdílet jej pomocí Samby či AFS.

Evidence SW, tedy přehled o tom, na kolika stanicích je aplikace nainstalována, kolik uživatelů ji smí používat, kde jsou instalační média, data pořízení, pořizovací ceny atp. Pro tuto oblast je dostupných na systému MS Windows mnoho komerčních řešení, které dokáží automaticky zjišťovat, jaký SW je na stanicích nainstalovaný. Dokáží i generovat přehledné výstupy a předávací protokoly. Tyto aplikace disponují mnohými funkcemi a možnostmi, bohužel v tom hlavním, tedy zjištění, jaké aplikace se na stanicích používají nezřídka selhávají. Je to dáno tím, že většina z těchto aplikací získává informace jen o SW, který byl

na stanici přímo nainstalován. U aplikací, které jsou spouštěny ze sdíleného disku na serveru a nezaněchávají konfigurační soubory na disku nebo v registrech, v detekci selhávají. I přes tento nedostatek mohou být, obzvláště s funkcí evidence počítačů, dobrým pomocníkem. Několik zástupců z této oblasti viz Tabulka 8.

http://www.alc.cz/ep/default.aspx	Systém evidence počítačů firmy ALC
http://www.auditpro.cz/cs/auditpro/	Evidence SW a HW firmy AuditPro
http://www.awcaesar.cz/	Evidence SW a HW firmy Free RW Soft
http://www.pcinfo.biz/article.php?id=27&lng=1	Modul evidence z balíku PCinfo fy. Fair Net

Tabulka 8: SW pro evidenci aplikací

2.3.4 Příprava instalace a repackaging

Pro zkoušení aplikací je dobré mít vyčleněn jeden výkonově průměrný počítač, s nejběžnějším operačním systémem v organizaci. K tomuto počítači je vhodné mít vyrobeno několik záloh systémové oblasti (image partition), případně mít k dispozici jiné mechanismy, jak mít software počítače pevně pod kontrolou. K vytváření a práci s image mohou doporučit následující programy – viz Tabulka 9.

http://www.partimage.org/	Partimage, Linuxový nástroj pro zálohu a obnovu partitions
Užitečné postupy < http://www.root.cz/clanky/klonovani-pocitacu/ >	
http://en.wikipedia.org/wiki/Dd_(Unix)	standardní utilita pro nízkoúrovňové
http://www.garloff.de/kurt/linux/ddrescue/	kopírování a její vylepšená varianta
http://www.chrysocome.net/dd	dd pro MS Windows > 9x
Norton Ghost – proslulý zálohovací nástroj partitions < http://www.symantec.com/home_homeoffice/downloads/index.jsp >	

Tabulka 9: aplikace pro zálohu a obnovu diskových oblastí

Máme-li více záloh systému v různém stádiu připravenosti stanice, lze testovat nový program v těchto odlišných prostředích. Například: Image1 po základní instalaci systému, Image2 po instalaci základních ovladačů HW, Image3 po instalaci oprav, rozšiřujících knihoven a základních programů, Image4 po instalaci veškerého běžného vybavení.

Soubory s image stanic mohou být k užítku po celou dobu existence stanice v organizaci. Vyplatí se vést si o nich evidenci, aby bylo snadné dohledat, co v kterém image je.

Zběžné zkoušení programů je možné provádět bez větších příprav, dokud systém počítače funguje. Na test aplikace, která má zapadnout do SW fondu organizace, je třeba se připravit důkladněji, tím spíše, pokud má být nasazena na velkém množství počítačů.

V tom případě je vhodné obnovit stanici do Image4, spustit monitoring registrů (MS Windows) případně zazálohovat stav adresáře s konfiguracemi (/etc/ Linux), provést zálohu stavu filesystému. Následně provést pokusnou instalaci a zaznamenat do logu rozdíly mezi současným a předchozím stavem. Poté otestovat fungování aplikace a opět udělat log z rozdílů oproti předchozímu stavu.

Tímto postupem budeme mít k dispozici dvě velmi důležité informace: 1. Co vše aplikace při instalaci dělá. 2. Co dělá při běžném užívání. Je dobré si uvědomit, že spuštěný program může dělat s předchozím SW cokoli, na co má uživatel, který jej spustil, práva. Nezřídka například právě pro instalaci, si programy vyžádají i práva administrátora. Bezostyšně běžný je tento přístup v systémech MS Windows. V tom případě může program v průběhu instalace dělat téměř cokoli. Jediný program tak může snadno vyřadit celý pečlivě budovaný systém. Tím, že máte k dispozici záznam o změnách provedených při instalaci, můžete alespoň zjistit, co vše program při instalaci dělá, případně co po něm bude potřeba opravit.

Poměrně často se můžete setkat s tím, že programy při instalaci mění bez dovolení asociace k typům souborů tak, aby soubory určitých typů byly otevírány jimi samotnými. Bez vyžádání vytváří ikony na ploše a v nabídce start, rozhazují své knihovny soubory po disku, vytváří si konfigurační soubory v systémovém adresáři, vytváří adresáře s uživatelským nastavením v domovském adresáři uživatele a společném adresáři všech uživatelů. Vytváří si nové klíče v registrech, upravují systémové konfigurační soubory, upravují registry tak, aby byly spouštěny automaticky po startu. Některé programy se chovají mnohem hůř. Při instalaci přepisují již existující dll knihovny vlastními verzemi, mění údaje v registrech a konfiguračních souborech, prohledávají obsah disku, ukládají a odesílají nežádoucí informace. Chování těch nejhorších hraničí s chováním počítačových virů. Například vytváří tzv. rootkity [4].

Záznam změn po použití aplikace může pomoci odhalit, jaké vazby má program na systém a prostředí. Může například pomoci detekovat místa, kam program ukládá registrační čísla, což může být užitečné, pokud chceme daný SW, bez nutnosti specializovaných utilit, automatizovaně instalovat na větší množství stanic.

Některé aplikace dokážou díky chybám nebo neshodám s prostředím samy sebe poškodit, či znefunkčnit. Může to být způsobeno chybami v konfiguračních souborech. Užitečné je mít k dispozici samostatnou zálohu všech výchozích souborů, případně i klíčů v registrech, do kterých program zapisuje. Nástroje pro záznam změn v systému, záznam činnosti programu a jejich analýzu viz Tabulka 10.

Online sledování přístupu k souborům, registrům, procesům < http://www.microsoft.com/technet/sysinternals/SystemInformation/Filemon.msp > < http://www.microsoft.com/technet/sysinternals/ProcessesAndThreads/Regmon.msp > < http://www.microsoft.com/technet/sysinternals/utilities/processmonitor.msp >	
http://asic-linux.com.mx/~izto/checkinstall/	CheckInstall pro Linux
http://www.epsilonsquared.com/installwatch.htm	InstallWatch pro MS Windows
InCtrl5 – precizní sledování aktivit programu, pro MS Windows < http://www.devhood.com/Tools/tool_details.aspx?tool_id=432 >	
UndoReg – Pro MS Windows – odstraňuje klíče a hodnoty v registrech, umí spolupracovat s výstupy z InCtrl5 a dalšími < http://publicdata.home.comcast.net/peplugins.htm >	

Tabulka 10: nástroje pro záznam změn v systému

Jako alternativní řešení k udržování konfigurace lze používat nástroje, které zaznamenávají stav systému a lépe či hůře dokážou hlídat a obnovovat změny souborů či registrů. Jde například o program Obnovení systému, který je automaticky k dispozici v OS MS Windows XP. Nevýhodou těchto programů je, že buď aktivně sledují a zaznamenávají změny, případně ukládají zálohy registrů a systémových souborů do vlastního archivu, což sebou nese snížení výkonu počítače a velké prostorové nároky na ukládání záloh. Ani pak nelze bezpečně říci, že bylo obnoveno vše potřebné.

Příprava na instalaci, obzvláště v prostředí open source operačních systémů, může zahrnovat fázi konfigurace překladače a překlad programu. K většině SW je dodáván i script, který tuto fázi řídí viz Tabulka 11. Kromě klasického Makefile či Antu k tomu mohou sloužit další pomocné scripty, případně bývá část přípravy a překladu popsána v dokumentaci instalace. U komplexnějších programů lze najít v dokumentaci další volby (direktivy), kterými se řídí kompilace. Jejich využitím lze docílit výrazných změn v možnostech a chování výsledné aplikace nově sestavené aplikace.

Makefile – program, který dle závislostí uvedených v popisném souboru, řídí přípravu a překlad zdrojových kódů < http://www.linux.cz/noviny/1999-0304/clanek12.html > < http://kengine.sourceforge.net/tutorial/g/makefile.htm >	
Ant – javovský nástroj pro kompilaci a komletaci dat < http://nb.vse.cz/~zelenyj/it380/eseje/xzoul01/a_ant.htm >	

Tabulka 11: nástroje pro řízení kompilace

K přípravě na instalaci se vztahuje velmi důležitý pojem, „repackaging“, neboli proces přebalení instalačního souboru do jiného typu instalačního souboru, a jak se tvrdí v knize [5], tři z pěti instalovaných programů, které mají vlastní instalátor, bývá přebaleno do některé obecnější verze instalátoru. Hlavní důvody bývají: potlačení uživatelských vstupů během instalačního procesu, ošetření chyb a problémového chování v cílovém prostředí, předpřipravení uživatelského nastavení.

Zde OS Linux mírně doplácí na své vysoce homogenní prostředí, protože různé distribuce mají různé instalátory. Ty využívají různé formáty pro uložení zdrojových nebo binárních souborů. I zde však existuje možnost přebalení instalačního balíčku z jednoho typu na jiný viz Tabulka 12.

Zabalení do jiného instalátoru se nemusí nutně týkat jen klasických aplikací. Jde zabalit i celý operační systém. S trochou nadsázky se dá říci, že i jednotlivé distribuce různých open source systémů jsou jakoby gigantickým repackem. Několik užitečných nástrojů pro provádění repackingu, viz Tabulka 12., a konverzí Tabulka 18.

http://www.nliteos.com/nlite.html	NLite, úprava instalátoru MS Windows
http://www.epsilonsquared.com/installrite.htm	InstallRite, repackaging tool
Propracovaný komerční nástroj Prism Deploy < http://www.newboundary.com/products/prismdeploy/ >	
http://autopackage.org/index.html	Pro Linux, multidistribuční balíčkovací framework
http://kitenet.net/~joey/code/alien/	Konverze mezi formáty balíčků rpm, dpkg, stampede slp, slackware tgz.
http://www.rpm2tgz.com/	Online převod z rpm na tgz formát

Tabulka 12: nástroje pro repackaging

V některých případech instalátor vyžaduje pro instalaci či běžné používání distribuční médium v mechanice. Dopad tohoto nepříjemného chování lze zmírnit použitím virtuálních mechanik viz Tabulka 13. Obraz virtuálního disku, obvykle typu ISO, lze vytvořit pomocí většiny vypalovacích programů. Scriptem pak lze zajistit automatické vložení virtuálního disku do virtuální mechaniky při spouštění aplikace. V prostředí Linuxu lze použít standardní program mount, který může namapovat i obraz instalačního média, jako by to byl disk.

http://www.daemon-tools.cc/	Daemon tools
http://www.alcohol-soft.com/	Alcohol 120%
http://www.elby.ch/products/clone_dvd/	CloneDrive
http://www.ezbsystems.com/ultraiso/	Editace ISO obrazu disku pro virtuální mechaniky

Tabulka 13: virtuální mechaniky

2.3.5 Instalace

Pokud je vše dobře připraveno, samotná instalace by měla být rychlá a snadná, neměla by vyžadovat vaši přítomnost u stanice a hlavně by měla být úspěšná. Za úspěšnou instalaci lze považovat tu, která umožní provozování aplikace na všech stanicích, na kterých aplikace má a může fungovat, aniž je zároveň negativně ovlivněn systém nebo jiné aplikace. Také

by spolu s instalací neměla proběhnout změna, která by vytvořila nevhodné prostředí pro budoucí aplikace. To je na první pohled obtížné odhadnout, ale dobrým vodítkem bývají kromě údajů z rozdílových logů také nároky, které má aplikace pro své fungování.

Pozitivně hodnotím aplikace, které bez problémů fungují, i pokud se umístí na sdílený disk na serveru s právem pouze ke čtení (read only), který je připojen (namapován / namountován) na klientských stanicích. Taková aplikace je mnohem více pod kontrolou, protože existuje jen jedna pracovní kopie, kterou navíc žádná stanice nemůže změnit. Aby aplikace mohla v takovém prostředí fungovat, nesmí být závislá na provádění akcí, které jí takové umístění nedovolují provozovat. Vhodné je, aby již od začátku byla pro takový způsob použití navržena a naprogramována. Mezi programy, které lze takto instalovat, se nachází i populární aplikace ze světa open source: kancelářský balík OpenOffice, prohlížeč Firefox, poštovní klient Thunderbird, grafický editor Gimp. I zde jsou rozdíly. Některé programy si ke svému fungování vystačí pouze s operační pamětí, některé, jako zrovna OpenOffice a Firefox, si založí adresář v domovském adresáři, kam ukládají nastavení vztahující se ke konkrétnímu uživateli. Je ale třeba počítat s tím, že ani aplikaci na read only disku, pokud je na stanici spuštěna, nebrání nic v činnostech, které umožňují přístupová práva uživatele. Do této kategorie bych ještě zařadil webové aplikace, které jsou v podstatě většinou také na read only disku. U nich bývá problémové chování automaticky lépe hlídáno.

Druhou skupinou jsou aplikace, které z nějakého důvodu potřebují mít možnost zapisovat do adresáře v němž se nacházejí. V prostředí Linuxu to není příliš běžné, zde programátoři využívají konvencemi vymezeného adresáře /var/ nebo /etc/, v prostředí MS Windows je to poměrně běžné. Zřejmě jde o přežitek z dob OS MS DOS. I tyto aplikace je vhodné dodatečně ošetřit read only přístupem, alespoň pro soubory, které právo zápisu opravdu nepotřebují. Takřka nezbytným krokem je záloha všeho, co je možné přepsat. Zajímavou možností je zneviditelnění, nebo znemožnění běžného procházení adresářů tak, že soubory uvnitř jsou sice dostupné, ale pouze přímým požadavkem, nikoliv výpisem obsahu adresáře. Bohužel některé programy to dokáže zmást natolik, že nejsou schopné pracovat. Toto chování lze simulovat například v souborovém serveru Samba nastavením vlastnosti „browseable = no”. V každém případě, pokud máme adresáře či soubory, které jsou otevřeny pro simultánní zápis více uživatelů nebo procesů, dostáváme se do problematiky kritických sekcí, kterou je potřeba řešit. Jak to řeší například souborový server Samba, je možné nahlédnout v 5. kapitole knihy [6].

Za poznámku stojí zajímavý fakt, že u starších počítačů připojených k rychlé síti může být rychlost přenosu dat ze sítě vyšší, než přenos dat z vlastního pevného disku.

Třetí skupina SW je nevyhnutelně instalována přímo na koncové stanice. Důvody jsou různé. Například jde o SW, který je speciálně určen dané stanici, jako například ovladače a obslužné programy. Také může být důvodem mobilita stanice, obvykle přenosného

počítače, kdy uživatel potřebuje mít k dispozici SW i mimo prostředí firemní sítě. Také může jít o tak strukturovanou aplikaci, a natolik využívanou, že její umístění mimo lokální stanici by nadměrně zatěžovalo komunikační linky.

Zatímco u aplikací umístěných na serveru se v mnohých případech lze skutečně instalaci vyhnout, u aplikací umístěných na stanici je nutné provést minimálně fázi zkopírování souborů. V prostředí běžných uživatelů nebývá vhodné, pokud instalace vyžaduje nějakou komunikaci s uživatelem. Nehledě na to, že mnoho instalátorů nemluví česky. Jak přizpůsobit instalátor svým požadavkům bylo již popsáno v kapitole 2.3.4. Další množnosti nabízejí různá komplexní řešení, která budou zmíněna v samostatné kapitole.

Někdy je i přes veškerou snahu o zautomatizování přece jen nutná vaše přítomnost u stanice. Poslední záchranou může být možnost přistupovat k ovládání stanice vzdáleně. Může být potřeba takovou stanici vzdáleně spustit, k čemuž slouží služba probuzení po síti wake on lan. Několik programů k doporučení viz Tabulka 14.

Putty – Telnet a SSH terminál pro prostředí MS Windows < http://www.chiark.greenend.org.uk/~sgtatham/putty/ >	
Funkce Vzdálená plocha – systémový nástroj MS Windows < http://www.microsoft.com/cze/windows/xp/pro/using/howto/gomobile/remotedesktop/default.msp? >	
http://www.realvnc.com/	RealVNC – server a klient vzdáleného přístupu
http://sourceforge.net/projects/teachercp/	Teacher Control Panel – sledování více stanic s VNC najednou
http://www.depicus.com/wake-on-lan/	Několik programů pro spouštění počítačů po síti z prostředí MS Windows
ether-wake – utilita pro wake on lan v prostředí Linuxu < http://packages.debian.org/stable/net/etherwake >	

Tabulka 14: programy pro vzdálenou správu

I přesto, že oficiální zdroje Microsoftu tvrdí: „*Správa je nejsnazší tehdy, když je v počítači používaném k provádění správy stejný operační systém jako ve vzdáleně spravovaném počítači.*“, [7] není třeba se obávat, že ke vzdálené správě serveru budete potřebovat ještě jeden.

2.3.6 Konfigurace

Existence nebo dostupnost aplikace v počítači ještě nemusí znamenat, že aplikace je připravena k používání, nebo že se chová tak, jak bychom potřebovali. Konfigurace je přizpůsobení chování aplikace našim potřebám nebo potřebám aplikace v daném prostředí.

U skupiny serverových aplikací, jakou je například Samba server, Apache, Tomcat, Ftp server, Kerberos, je správná konfigurace mnohonásobně náročnější než samotná instalace. Většina těchto aplikací není bez správné konfigurace vůbec funkčních. Rozumět a umět tyto aplikace správně nastavit bývá často jedním z hlavních úkolů administrátorů. Porozumění a zvládnutí konfigurace některých programů může být i několikaměsíční práce.

Většina uživatelských programů takové nároky nemá a je možné je nastavit poměrně snadno. Tedy jeden jde nastavit snadno. Pokud máte provést jednu snadnou instalaci na 300 počítačích, snadnost se rychle vytrácí. Někdy jedna stanice slouží více uživatelům a program je třeba nastavit pro každého z nich. A to už je opravdu hodně snadných instalací najednou.

Část konfigurace se dá umístit do přepracovaného instalačního balíčku. Část se dá provést po instalaci pomocí scriptu, který instalaci řídí. Poměrně běžná bývá konfigurace, která probíhá při přihlášení uživatele, tzv. login scripty. Někdy je třeba provádět konfiguraci při každém spuštění nebo ukončování programu, v tom případě je třeba se „naháčkovat“ (hook) na proces spouštění. Zní to složitě, ale v podstatě jde jen o to, že uživatel nespouští přímo aplikaci, ale váš script, který může něco vykonat před samotným spuštěním. Následně spustí aplikaci, a když aplikace končí, řízení může být předáno zpět scriptu, který může dodatečně něco vykonat. Pro zmenšení míry matení uživatele bývá vhodné, aby zástupce, kterým uživatel script spouští, měl stejné jméno i ikonu jako samotný program. Několik zástupců scriptovacích programů viz Tabulka 15.

Co je to vlastně konfigurace? Jsou to nastavení a volby, které určují chování, případně i vzhled programu. Patří sem tedy například rozložení nástrojových ikon v textovém editoru, rozložení ikon na ploše, historie zadaných příkazů v Linuxovém shellu [8], bezpečnostní nastavení prohlížeče webových stránek. Patří sem ovšem i komplexnější data, jako například uživatelský slovník, zapamatovaná hesla pro přístup na webové stránky, pluginy různých programů a jejich nastavení a mnoho dalšího.

Konfigurace patří mezi velmi těžko obnovitelná data. Podobně jako dokumenty vytvořené uživatelem nebo save soubory počítačových her. Podle toho bychom s nimi měli zacházet. Tedy zálohovat. Rozdílové zálohy vám také mohou hodně pomoci při detekci nových problémů s chováním aplikací. Často změny chování aplikací mají svůj otisk, ne-li přímo příčinu v konfiguračních souborech.

BATch soubory vykonávané interpretem příkazové řádky command Fy Microsoft < http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/batch.msp?mfr=true >	
Rozcestník scriptování pro MS Windows < http://www.microsoft.com/technet/scriptcenter/createit.msp >	
BASH – tradiční shell v Linuxu a jeho scriptovací možnosti < http://docs.linux.cz/programming/interpreted/bashdoc-1.4/ >	
http://ipython.scipy.org/moin/	IPython – shell jazyka Python se zaměřením na podporu systémového scriptování

Tabulka 15: scriptovací nástroje

Zatímco na správu konfigurací uloženou v textových souborech lze používat mnoho běžných nástrojů, s konfigurací uloženou v registrech MS Windows je práce o řád obtížnější. Použít lze specializované nástroje, například regedit.

2.3.7 Zpřístupnění a znepřístupnění aplikace

Aplikace je dostupná a může být na stanici provozována, ale z nějakého důvodu, ať už organizačního nebo licenčního, si nepřejeme, aby konkrétní uživatel nebo všichni uživatelé konkrétní stanice nemohli aplikaci používat. Ve speciálních případech můžeme dostupnost aplikací omezovat na určitý časový úsek, nebo podle dalších kritérií.

Velmi jednoduchý, přesto často dostačující způsob znepřístupnění aplikace, je odstranění zástupce, kterým uživatel aplikaci spouští. Propracovanější metody vedou ke skutečnému znepřístupnění aplikace. K tomu účelu lze využít již zmíněné login scripty, práva uživatelů či skupin, nebo jejich kombinaci. Některé aplikace se snaží tuto oblast řešit po svém a obsahují vlastní vrstvu pro řízení přístupu.

2.3.8 Výběr a spouštění

Tato část je ve většině případů v grafickém prostředí redukována na prosté spuštění aplikace prostřednictvím zástupce na ploše, případně v nabídce programů. Znalosti o adresářové struktuře jsou u většiny běžných uživatelů redukovány na adresář s vlastními dokumenty. Obecnější povědomí o principech adresářové struktury, o spouštěcích souborech a zástupcích chybí. Nelze jednoznačně říci, že je to špatný stav. Pokud tyto znalosti ke své práci nepotřebují a nevyužili by je, není ani příliš důvodu je tímto zatěžovat.

Tomuto tématu je věnována samostatná kapitola 4.5. na straně 48.

2.3.9 Používání aplikací

Nejdůležitější část života aplikací. Tato fáze je doménou uživatelů, pro které je aplikace určena. I když uživatelé předpokládají, že administrátor zná dopodrobna funkce všech programů, nemusí tomu tak být. Do kategorie systému správy SW však z této oblasti patří situace, kdy aplikace přestane správně fungovat.

Postupy i nástroje pro řešení problémů při fungování aplikací jsou různorodé. Závisí především na způsobu a četnosti projevování se chyb a jejich volba bývá závislá na zkušenostech administrátora. Obecně je doporučováno postupovat od nejméně náročných kroků ke složitějším. Pokud problémy setrvávají i po restartu stanice, bývá třeba detekovat, kde chyba nastává. Zda je způsobena přímo aplikací, špatnou konfigurací, interakcí s prostředím, zda jde o chybu vnějšího prostředí, ve kterém aplikace pracuje, případně zda

jde o chybu HW. Hodně může napomoci zjištění posledních akcí provedených s aplikací, systémem nebo stanicí. Často bývá příčinou poruch něco, co se událo mezi stavem, kdy aplikace fungovala bezchybně a kdy se objevily problémy.

Jednou z diagnostických technik je zjistit, co aplikace dělá, než se chyba projeví. K tomu lze použít stejné nástroje, jako v případě sledování instalace viz Tabulka 10. Další nástroje, viz Tabulka 16.

http://support.microsoft.com/kb/323527/cs	Správce úloh MS Windows (karta procesy, menu zobrazit sloupce, výběr položek)
Netstat – MS Windows, sledování síťové aktivity < http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/netstat.msp?mfr=true >	
http://www.computerhope.com/unix/unetstat.htm	Netstat v Linuxu
http://www.linuxzone.cz/index.phtml?ids=9&idc=279	Popis několika utilit pro sledování systému v Linuxu
http://www.tiler.com/freemeter/	FreeMeter – MS Windows sleduje několik systémových oblastí
http://www.ethereal.com/	Ethereal – multiplatformní program, podrobný rozbor síťové komunikace
http://kobik.net/spy_capture.asp	Spy & Capture – rozbor vnitřní struktury aplikace v prostředí MS Windows

Tabulka 16: nástroje pro sledování a ovlivnění běhu aplikací

Pro plánování rozvoje a údržby i pro přehled o systému je užitečné mít k dispozici statistiku o tom, které programy uživatelé používají. Získávání těchto údajů je citlivá záležitost, kterou uživatelé vnímají jako zasahování do soukromí. Může to však být, spolu s údajem o zatížení klíčových komponent, jeden z hlavních údajů zpětné vazby o fungování a využívání systému. Jsou-li data zjišťována anonymně, je to pro uživatele přijatelnější.

2.3.10 Opravy SW, reinstalace

Když aplikace vykazuje chybné chování a příčinou tohoto chování je poškození její části, pak oprava znamená vrácení části aplikace nebo její konfigurace do předchozí podoby, kdy nefungovala chybně. Přitom je potřeba, pokud možno, neměnit uživatelská data a uživatelské nastavení. U vhodně navržené aplikace by mělo být snadné uvést ji do výchozího stavu, doplnit ji konfigurací a uživatelskými daty. Touto funkcí disponují i některé instalátory, včetně standardního MSI.

Pokud je chyba způsobena špatným zpracováním uživatelských dat nebo konfigurace, bývá oprava obtížnější. Pomoci může analýza dřívější zálohy, případně srovnání s funkční verzí na jiné stanici.

V případě, že, jde o vnitřní chybu programu, která se jen dosud neprojevila, je někdy možné vyhnout se rizikovému sledu akcí, které způsobí selhání. Mělo by to však být jen dočasné řešení, než výrobce SW vydá opravný balíček (patch), nebo novou verzi bez této chyby. Pro MS Windows vychází opravy, především chyb ohrožujících bezpečnost systému, poměrně pravidelně každý měsíc. U mnohých výrobců je velmi obtížné domoci se opravy a to i po zajištění detailní analýzy chybového chování. V prostředí open source jsou opravy dostupné, jakmile jsou vyrobeny a otestovány. To může být za několik hodin od zjištění, případně v nedohlednu, nemá-li již SW své vývojáře. Podstatou je, že jsou-li k dispozici zdrojové kódy, je možné v nich opravu přímo provést. Nebo si lze najmout někoho, kdo to udělá za vás. Logickým krokem by pak mělo být zveřejnění takové opravy pro ostatní.

Existuje i přístup, který se snaží vrátit operační systém a nastavení aplikací do základního stavu při každém spuštění stanice. Tento postup se používá u informačních kiosků a tam, kde je důležité zachovávat jednotné prostředí. Tento postup má své negativní stránky, které se začnou projevovat, je-li na stanicích potřeba častěji měnit konfiguraci nebo aplikovat patche. Komplexním řešením je například aplikace firmy SodatSW, OptimAccess, která je v Čechách poměrně rozšířená. Je určena pouze pro prostředí MS Windows. „*Systém OptimAccess chrání nastavení instalovaných programů, operačního systému, komunikačních prostředků a nastavení Windows na pevných discích jednotlivých počítačů.*”
<<http://www.pachner.cz/html/tipy/optimaccess.htm>>.

Krajním řešením, které nemusí přinést úspěch bývá reinstalace. Aplikace se nejdříve standardním způsobem odinstaluje a poté se nainstaluje znovu. Pokud je chyba zapříčiněna vadou konfigurace nebo interakcí aplikace s prostředím, může být aplikace po reinstalaci stále problémová.

2.3.11 Migrace aplikace

Přesun aplikace. Může to být přesun do jiného adresáře nebo na jiný disk. Může to být přesun do jiné sítě, k jinému uživateli, přesunutí do nového operačního systému. To vše může být v rámci jedné stanice nebo může jít o přesun na jinou stanicí.

Migrace není prosté zkopírování souborů. Jde o docílení stavu ve výchozím prostředí, jakoby aplikace vůbec nebyla nainstalována, případně byla odinstalována. Následně pak stavu cílového prostředí, jako by zde aplikace byla již od začátku jejího používání. Přenáší se tedy nejen soubory samotné aplikace, ale také její konfigurace. Mění se vazby na prostředí. Například asociace souborů, kontextová menu a pod.

Asi nejčastějším typem je migrace aplikací ze stávajícího operačního systému do nového. Jednou z možností je instalovat nový operační systém způsobem upgrade stávajícího, což může, ale nemusí, být funkční postup. Lepších výsledků může být dosaženo zabalením současných aplikací do instalačních balíčků a jejich instalace v novém systému. Této problematice se věnuje kniha [9].

Z komerčních řešení zmíním firmu Versora <<http://www.versora.com/>>, která se specializuje na migraci uživatelských dat nejen mezi verzemi jednoho typu OS, ale také mezi MS Windows a Linux. Oficiálně je podporováno 12 Linuxových distribucí, ale jejich počet není tímto číslem omezen.

2.3.12 Upgrade, aktualizace a doplňky

Pomineme-li opravné patche zmíněné v kapitole o opravách, pak upgrade má za úkol doplnit stávající aplikaci o novou funkčnost. Některé upgrade se řeší odinstalováním původní verze a nainstalováním nové s tím, že konfigurace a uživatelská data jsou zachována. Nejčastěji jsou v průběhu upgrade přepisovány některé původní soubory novými verzemi. Někdy jsou aktualizace brány systémem jako klasická instalace, například v případě upgrade MS Windows, někdy nejsou instalátorem vůbec zachyceny.

Spolu s vývojem v oblasti komunikačních sítí se rozvíjí nové modely vývoje a distribuce SW. Nadvláda tzv. krabicového SW se přesunula k SW s podporou poskytovanou po Internetu a pomalu se svažuje k poskytování pronajatých služeb.

Frekventované skupiny SW, jakou jsou antiviry, antispyware, antispamy i některé firewally, mají časté aktualizace jako jeden z hlavních požadavků. Tento druh aktualizací většinou nemění mnoho ze současné aplikace. Bývá to jeden, nebo několik málo souborů s indexy aktualizací. Většinou přidává data do vyčleněného adresáře. Převážně jsou to data o nových virech a spyware. Je běžné nastavovat stanice na automatickou aktualizaci, což jistě pomáhá proti šíření nových virů a spyware. Je třeba počítat s riziky v případě, že aktualizace samotná omylem nebo záměrně vykoná něco nežádoucího.

Doplňky (add-ons) jsou pojmem, který se do popředí dostal spolu s rostoucí popularitou webového prohlížeče Mozilla Firefox. Add-ons představuje souhrnný název pro rozšíření (extensions), zásuvné moduly (plugins), motivy vzhledu (skins), slovníky a další SW celky, které doplňují funkcionalitu základní aplikace. Role mezi základní aplikací a doplňky je podobná roli operačního systému a aplikací. V základní aplikaci bývá komponenta pro správu doplňků. Doplňky lze instalovat, konfigurovat, aktualizovat, aktivovat a deaktivovat, mazat.

Možnost rozšiřovat aplikaci pomocí doplňků má mnoho kladných rysů. Aplikace může obsahovat ty funkce, které uživatel opravdu potřebuje. Vývojáři aplikace se mohou soustředit na základní aplikaci a její podporu pro doplňky. Struktura aplikace bývá jednodušší a snáze se hledají a odstraňují chyby. Velkou nevýhodou doplňků je jejich izolovanost od standardních prostředků pro řízení instalace. Nezřídka je celý kód doplňku ukládán do profilu uživatele, konfigurace rozšíření nebývá nabídnuta ke smazání.

2.3.13 Odinstalace

Když pominou potřeby, které stály za vznikem aplikace, jejím nasazením nebo rozhodnutím, že tyto potřeby bude splňovat jiná aplikace, přistupuje se k odinstalování. Pokud nejsou žádné důvody pro okamžitou odinstalaci, není třeba tento krok uspěchat.

Po odinstalaci se chce, aby ze systému byly odstraněny všechny soubory, které do něj aplikace pro svou potřebu přidala, vyjma těch, které mezitím začaly potřebovat jiné aplikace. Dále aby bylo možné nastavit, zda se má odinstalovat i uživatelská konfigurace. U MS Windows – aby z registrů byly odstraněny všechny klíče, které zde aplikace přidala a sloužily pouze k její potřebě.

Jen zřídka lze najít aplikaci, která se dovede správně odinstalovat. V prostředí MS Windows existuje kromě standardní systémové utility „Přidat nebo odebrat programy“ velké množství aplikací, které odinstalaci zajišťují, viz Tabulka 17. Některé se specializují jen na konkrétní aplikace, např. DirectX 9 Uninstaller, který má na serveru www.slunecnice.cz téměř 30 000 stažení (březen 2007). Do kategorie specializovaných odinstalátorů by se daly zařadit i některé aplikace jednorázově odstraňující viry či spyware.

http://www.martau.com/index.php	Total Uninstall
http://www.zsoft.dk/index/software_details/4	ZSoft Uninstaller
http://support.microsoft.com/kb/290301	Windows Installer CleanUp – uklízí nepodařené odinstalační údaje

Tabulka 17: odinstalátory aplikací v MS Windows

3 Existující komplexní řešení správy SW

Aktivít snažících se o zjednodušení správy SW je mnoho. Ucelené přístupy a postupy, které by postihovaly celou problematiku správy SW, již existují. Bohužel jen obtížně pronikají k nasazení v malých a středních organizacích.

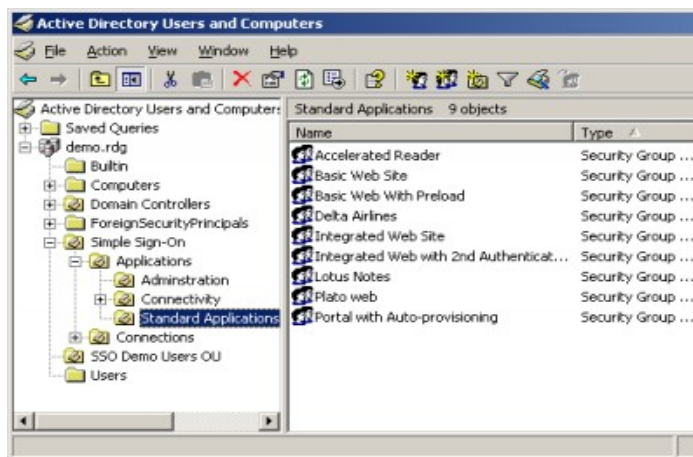
V předchozích kapitolách je vidět, že systém správy SW je poměrně rozsáhlý. Ironií je, že celý tento proces, který se odehrává mezi programátorem a uživatelem, je zde, takřkajíc, navíc. Je z velké části způsoben nedokonalostmi, jak na straně tvůrců koncepcí, programátorů a distribuce, tak i na straně tvůrců operačních systémů. Řada SW firem se snaží záměrnou nestandardností vytvořit technologickou závislost uživatelů na svých produktech. (Podobně tomu bylo ještě v 60. letech s HW). Například operační systémy firmy Microsoft poskytují jen slabou podporu normy POSIX, standardizující základní služby OS, která existuje již 20 let.

Prozatím jsme se zabývali tím, jak pomocí utilit a vhodných postupů docílit zautomatizování, zjednodušení a usnadnění jednotlivých činností správy SW. V následujících kapitolách bude stručný přehled některých komplexních řešení, která jednotlivé operační systémy, distribuce, nebo další subjekty nabízí.

3.1 V prostředí OS MS Windows

3.1.1 Microsoft - Systems Management Server

Základem funkcí systému je poskytování a řízení přístupu ke zdrojům. Počítač, aplikace, uživatel, skupina, doména, podsítě jsou zde brány jako entity, které jsou spolu provázány ve stromové struktuře. Tento produkt je na trhu již od roku 1994. Spolu s vydáním OS Windows 2000 server se jeho klíčovou částí stala služba Active Directory a v OS Windows 2003 server byl tento systém výrazně doplněn o další funkcionalitu. Pro jednotlivce a menší skupiny je dostupná minimalizovaná verze ADAM (Active Directory Application Mode). Zavedení tohoto řešení nabízí u nás řada firem, např. AutoCont, reference v ČR/SR: E-ON, Kooperativa pojišťovna a.s., Mostecká uhelná a.s.,



Obr. 1: konfigurační prostředí Active Directory

Využití Active directory v prostředí českých školy popularizuje projekt Repair 2000<<http://www.repair2000.cz/>>

Obecné vlastnosti MS Systems Management Server				
Multiplatformní	I18N / čeština	Modulární	Konzole / GUI	Repozitář SW
ne ^{*1)}	ano / částečně	částečně	ne / ano	minimální ^{*2)}

1) lze docílit připojení zdrojů z jiných platforem a v omezené míře poskytovat některé služby

2) jen bezpečnostní update

Podrobnější informace: <<http://www.microsoft.com/smsserver/default.msp>>.

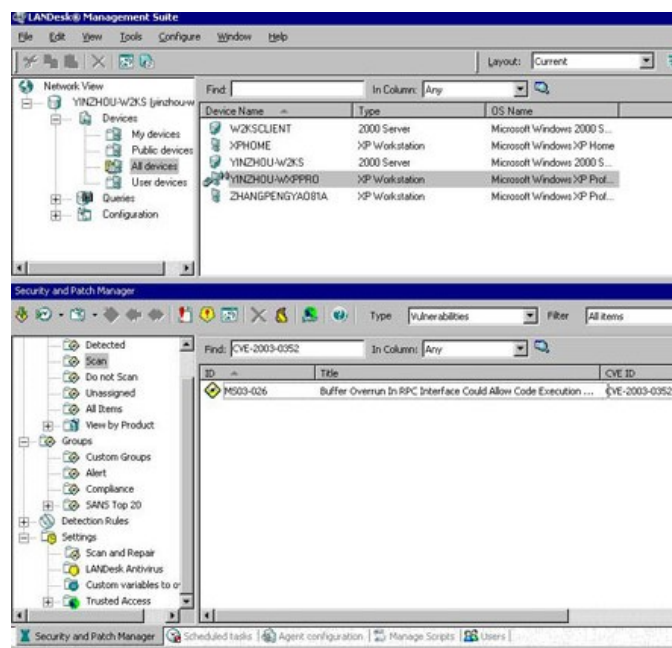
Kniha v čeština o active directory: <<http://www.grada.cz/katalog/kniha/active-directory/>>

3.1.2 Landesk



Spolupracující sada nástrojů s cílem pokrýt životní cyklus HW i SW v podniku. Základním nástrojem je Management Suite, dalšími pak Security Suite rozšiřující možnosti řízení přístupu a zabezpečení, Patch Manager centralizující správu bezpečnostních oprav, Asset Manager ekonomický přehled zdrojů, Server Manager řízení serverů a řešení v případě výpadku, System Manager monitorování a údržba systémů.

Systém se snaží podporovat i přenosné stanice, které pomocí tunelového spojení mohou komunikovat s centrálou organizace.



Obr. 2: konfigurační prostředí Landesk

Obecné vlastnosti Landesk				
Multiplatformní	I18N / čeština	Modulární	Konzole / GUI	Repozitář SW
ne ^{*1)}	ne / ne	ano ^{*2)}	ne / ano	ne

1) lze docílit připojení zdrojů z jiných platforem a v omezené míře poskytovat některé služby

2) jde spíše o spolupracující aplikace

Podrobnější informace: <<http://www.landesk.com/>>. V Čechách se o zavádění a servis stará například firma Clarionet <<http://www.clarionet.cz/landesk.html>>. Reference v ČR/SR: Česká exportní banka, a. s. Dopravní podnik města Brna, a. s. Ministerstvo pro místní rozvoj ČR, Vysoká škola chemicko–technologická v Praze.

3.1.3 Další řešení systému správy SW

Novell – eDirectory – <<http://www.novell.com/products/edirectory/>>

CA – Desktop Management Suite for Windows

<<http://www.ca.com/us/products/product.aspx?ID=4882>>

Oracle – Enterprise Manager s Enterprise Manager Application Server Control –

<http://www.oracle.com/enterprise_manager/index.html>

IBM – Tivoli Framework s IBM/Tivoli Configuration Manager –

<http://www-5.ibm.com/cz/software/tivoli/management_framework.html>

FairNet – PCinfo MagicEYE – <<http://www.pcinfo.biz/article.php?id=16&lng=1>>

BMC – Performance Manager (PATROL®) s Manager for Business Applications

<http://www.bmc.com/products/products_services_detail/0,,0_0_0_2001,00.html>

3.2 V prostředí OS Linux

Zpočátku bylo software pro Linux velmi málo. Vývoj se soustředil na jádro a základní aplikace. Byl téměř výhradně určen technicky zdatným jedincům, pro které ruční řešení závislostí, konfigurace, kompilace a instalace SW ze zdrojových kódů neměla být problémem.

Dnes je tomu výrazně jinak. Pro Linux jsou k dispozici tisíce aplikací a nabízí komfortní uživatelské prostředí. Rozšiřuje se od dříve typického nasazení v serverové oblasti k nasazení na desktopech běžných uživatelů. V některých oblastech systému správy exceluje. Nevýhodou je současná existence několika de-facto standardů pro distribuci SW. Existují však standardizační snahy o zavedení jednotného formátu a existuje řada nástrojů, které pro některé formáty dokáží provádět konverzi, viz Tabulka 18.

Mezi nejvýznamnější standardizační snahy v oblasti správy SW patří aktivity organizace Linux Standard Base (LSB) – <<http://www.linux-foundation.org/en/LSB>>.

http://kitenet.net/~joey/code/alien/	Alien – konverze mezi formáty rpm, deb, stampede slp, slackware tgz
http://www.rpm2tgz.com/	Online konverze rpm na tgz
http://freshmeat.net/projects/checkinstall/	CheckInstall – při instalaci ze zdrojových kódů simuluje instalaci rpm, deb, slackware
http://en.opensuse.org/Build_Service	Online překladová a balíčkovací služba
http://apt4rpm.sourceforge.net/	apt4rpm – konverze z rpm formátu na apt, yum, nebo balíček s metadaty

http://www.general.uwa.edu.au/u/toivo/kpackage/	Kpackage – multiformátový balíčkový manager RPM, Debian, Slackware, BSD package manager
makepkg – utilita pro tvorbu balíčků < http://www.hpl.hp.com/personal/Carl_Staelin/mkpkg/help/ >	
Standardní balíčkovací program distribuce Arch Linux < http://www.archlinux.org/pacman/makepkg.8.html >	
http://autopackage.org/index.html	Autopackage – multidistribuční balíčkovací framework

Tabulka 18: pomocné nástroje pro tvorbu balíčků a konvertory

K základním programům existuje mnoho nadstaveb, které rozšiřují možnosti práce s balíčky, nebo přidávají propracovanější uživatelské prostředí. Podrobnější přehled vlastností Linuxových distribucí, viz <http://en.wikipedia.org/wiki/Comparison_of_Linux_distributions>.

Na rozdíl od OS MS Windows obsahuje většina distribucí, kromě jádra systému a základních programů, velké množství otestovaných a připravených aplikací. Tyto aplikace jsou k dispozici okamžitě po instalaci, případně lze v průběhu instalace zvolit, které budou nainstalovány. Pro mnoho distribucí existují veřejně přístupné repozitáře SW, čítající i několik tisíc programů. Na počítači připojeném k Internetu lze pomocí managerů balíčků provést stažení, konfiguraci i instalaci jediným příkazem. U některých distribucí lze jediným příkazem provést i aktualizaci veškerých nainstalovaných programů, ke kterým byla vydána nová verze.

3.2.1 Založené na RPM

RPM je jednak název pro správce balíčků a jednak pro formát, ve kterém jsou programy distribuovány. Za vývojem rpm stojí firma Red Hat. Vzhledem k popularitě tohoto formátu se na nástrojích, které jej využívají, podílí řada dalších skupin. V základní podobě dokáže manager programy instalovat, odinstalovat, ověřovat pravost a poškození, vyhledávat v nich informace, aktualizovat.

Několik distribucí, které tento formát používají: Red Hat Enterprise Linux, the Fedora Project, SUSE Linux Enterprise, openSUSE, CentOS, Mandriva Linux. Formát rpm je doporučován i standardizační organizací Linux Standard Base.

http://rpm.org/max-rpm/ http://www.mikajik.info/rpm.html	rpm – základní nástroj, popis anglicky, popis česky
rpm -qa grep balik	za pomoci grep vyhledá program balik
rpm -i /home/uzivatel1/balik-2.9.rpm	provede instalaci programu balik 2.9
rpm -e balik	odinstaluje program balik
yum	manager instalací distribuce Fedora
urpmi	manager instalací distribuce Mandriva
YaST	manager instalací distribuce Suse
https://moin.conectiva.com.br/AptRpm	manager apt s podporou rpm
http://www.general.uwa.edu.au/u/toivo/kpackage/	Kpackage – grafická ná

Tabulka 19: nástroje pro práci s formátem rpm

3.2.2 Založené na DEB

Formát DEB je původem z distribuce Debian, nyní jej využívají mnohé další: Knoppix, Ubuntu a jeho klony, Xandros.

http://wiki.ubuntu.cz/dpkg	dpkg – základní nástroj
dpkg -l grep balik	za pomoci grep vyhledá program balik
dpkg -i /home/uzivatel1/balik-2.9.deb	provede instalaci programu balik 2.9
dpkg -r balik	odinstaluje program balik
apt - manager instalací distribuce Debian < http://www.debian.org/doc/manuals/apt-howto/index.cs.html >	
apt-cache search balik	vyhledá balíčky obsahující v popisu balik
apt-get install balik	nainstaluje program balik
apt-get remove balik	odinstaluje program balik
apt-get update	aktualizuje informace o programech
apt-get upgrade	provede aktualizaci aplikací
aptitude – manager instalací s rozšířeným ovládáním < http://people.debian.org/~dburrows/aptitude-doc/cs/index.html >	
http://wiki.ubuntu.cz/Synaptic	Synaptic – grafická nadstavba mana

Tabulka 20: nástroje pro práci s formátem deb

3.2.3 Specializované

Distribuce Linux Gentoo má jeden z nejpropracovanějších instalačních managerů Portage. Základní koncepce byla převzata ze systému BSD a rozšířena. Portage ve spolupráci s online repozitářem SW, ve kterém je (duben 2007) více než 11 000 balíčků, představuje

hlavní nástroj správy této distribuce. Obsahuje přes 350 obecných a 1200 specifických konfiguračních značek, kterými lze ovlivňovat průběh kompilace a tak i vlastnosti a chování instalovaných programů. Některé z vlastností systému správy Gentoo: automatické nastavení nejvhodnějších mirrorů pro získávání aplikací, optimalizovatelnost instalovaných programů pro konkrétní HW, aktualizace seznamu SW v repozitáři, manuální nebo automatický upgrade vybraných aplikací, nebo celého systému, automatické řešení závislostí před instalací, automatický download aplikace a všech závislých aplikací, vyhledávání v názvech programů a jejich popiscích, instalace podle několika vývojových větví aplikace, možnost instalace stejné aplikace ve více instancích, automatické a manuální řešení případných konfliktů v konfiguraci, virtuální jména služeb, které jsou provázány na konkrétní aplikaci, a další...

http://www.gentoo.org/	Oficiální stránky
http://gentoo-portage.com/	Oficiální stránky portage
http://www.fuzzy.cz/gentoo/files/html/ch15.html	Návod k nastavení rozšířených možností Portage
http://packages.gentoo.org/	Informace o dostupných balíčcích
http://www.gentoo.org/doc/cs/	Dokumentace k Gentoo v češtině
Návod pro práci s balíčkovacím systémem < http://www.abclinuxu.cz/clanky/navody/balickovaci-system-gentoo-linuxu-i >	

Tabulka 21: zdroje k distribuci Gentoo a programu Portage

Distribuce Slackware <<http://www.slackware.com/>> [10], patří mezi nejstarších distribuce. Pro instalace používá balíčky .tgz, což je balíček zabalený programem tar a komprimovaný programem gzip. Uvnitř balíčku bývají soubory s popisem, kam instalovat, co instalovat, popis závislostí a pod. Jako manager instalací zde funguje sada nástrojů souhrnně nazvaných pkgtools. Najdeme zde: installpkg, removepkg, upgradepkg, makepkg, a pkgtool. Dále nadstavby pro práci s repozitáři: konzolový SWARET a grafický GSlapt. Online repozitář balíčků lze najít na adrese <<http://packages.slackware.it/>>

Distribuce Slax <<http://www.slax.org/>> používá k instalaci programů mechanismus modulů. Moduly jsou uloženy v .img souborech. Příkazem moduse lze modul připojit k adresářové struktuře a program se tak stane nainstalovaným. Katalog modulů <<http://www.slax.org/modules.php>>

Distribuce Gobo Linux <<http://www.gobolinux.org/>> má k instalacím velmi radikální přístup. Na titulní stránce je výstižná věta „In GoboLinux you don't need a package manager because the filesystem is the package manager”. Každý program má vlastní adresář a neumísťuje žádné soubory mimo něj. V případě, že některý soubor je potřeba v jiném adresáři, pouze se na něj vytvoří link. Instalační balíčky jsou prosté .tar.bz2 zabalené adresáře

s programem, navíc obsahují pouze seznam, kam se mají vytvořit linky. Ke správě programů obsahuje i grafickou nadstavbu, Program Manager. Repozitář oficiálních balíčků je k nalezení na adrese <<http://gobo.kundor.org/packages/official/>>.

Fai – Fully Automatic Installation <<http://www.informatik.uni-koeln.de/fai/>>, sada shellovských a perlowských scriptů, které jsou připraveny pro zautomatizování kompletní přípravy a instalace stanice či serveru. Tento postup těží z ze silné stránky Linuxu, z příkazové řádky a podpory scriptování. Na rozdíl od činností v GUI lze postupy prováděné v příkazové řádce poměrně snadno nahradit scriptem. S doplněním o možnost využití programovacího prostředí shellu, perlu nebo jiných scriptovacích jazyků, lze tímto způsobem řešit i složité akce.

Univerzální řešení instalačních balíčků se snaží zavést několik skupin.

http://autopackage.org/index.html	AutoPackage – systém snažící se o univerzální balíčkovací formát
http://klik.atekon.de/	Klik – systém snažící se redukovat instalaci na pouhé zkopírování

Tabulka 22: Nástroje pro interdistribuční instalaci aplikací

3.3 V prostředí OS BSD

<<http://www.bsd.org/>> Nabízí se možnost tradiční instalace ze zdrojových kódů, nebo instalace binárních balíčků. Informace o SW je udržována v databázi pkgdb. Ta obsahuje přehled o všech dostupných a nainstalovaných balíčcích, jakož i jejich závislostech. Po zkompileování je program automaticky přidáván do seznamu dostupných aplikací, včetně uvedení závislostí.

pkg_info	výpis databáze aplikací z /var/db/pkg
pkg_add -r balik	nainstaluje program balik
pkg_delete -x balik	odinstaluje program balik
pkgdb -f	aktualizace databáze aplikací
cvsup & portsdb -Uu	aktualizuje strom aplikací
portupgrade	provede aktualizaci aplikací

Tabulka 23: správa SW v OS BSD

Klon BSD, distribuce PC-BSD <<http://www.pcbsd.org/>> se ve správě SW snaží přiblížit běžným uživatelům. Aplikace jsou distribuovány v balíčku spolu s knihovnami, na kterých jsou závislé. Instalace probíhá stylem Next-> Next->OK. Formátu balíčků lze rozpoznat podle koncovky .pbi. Typ souboru je asociován s instalátorem, instalátor tak má přehled o všech instalovaných programech. Repozitář balíčků <<http://www.pbidir.com/>>

3.4 V prostředí OS Solaris

OS Solaris <<http://www.sun.com/software/solaris/>> má, na rozdíl od Linuxu, konzistentnější způsob vývoje, který zajišťuje firma Sun Microsystems. V roce 2004 byl, jako projekt OpenSolaris, otevřen zdrojový kód i pro vývojáře mimo komunitu Sunu. Od té doby vzniklo již několik distribucí, hlavní směr však drží Sun s původní distribuční řadou. Na klíčové oblasti, jako je systém správy SW, jsou nasazovány komplexní řešení vysoké úrovně. Integrovan je aplikační server Sun Java™ System Application Server Platform Edition spolu s Sun ONE Directory Server pro řízení identity. Záměrem je taktéž vyšší podpora aplikací založených na technologiích Java, XML a SOAP.

Z dalších technologií pro správu SW je to především Software Solaris Flash, který umožňuje vytvořit obraz celé stanice nebo serveru a přenést jej na jiný. Change Manager pro podporu provádění upgrade nebo instalaci nového programu na všech stanicích a serverech v nadefinované skupině. Pomocí aplikace Sun Sigma přináší podporu konfigurací. Přináší pokročilou techniku softwarových kontejnerů, virtualizace, Dynamic Tracing k identifikaci problémů, vestavěné adresářové služby, dynamickou rekonfiguraci, Jumpstart umožnění rychlé obnovy systému, Java Desktop System integrovaná podpora klientských stanic a další.

Správa SW SysV packages, nástroj pkgadd.

3.5 V prostředí MAC OS

Mac OS <<http://www.apple.com/macosx/leopard/>> firmy Apple, v poslední verzi nesoucí označení X, disponuje instalátorem Mac OS X Installer. V dřívějších verzích byl instalátor k nalezení pod označením Apple Installer. Oba pracují s formátem souborů .pkg. Tento formát byl vyvinut firmou NeXT a později převzat firmou Apple Computer. K vytváření balíčků se používá nástroj PackageMaker.

Další možností je, umístit speciální soubor s vlastním typem .app, typicky do adresáře /Applications. Soubor obsahuje vlastní mini souborový systém. Instalací je jeho připojení. Další nástroje viz <http://en.wikipedia.org/wiki/Apple_Developer_Tools#PackageMaker>.

Popis správy SW je popsán v dokumentu:

<<http://developer.apple.com/documentation/DeveloperTools/Conceptual/SoftwareDistribution/index.html>>

Speciální instalátor Boot Camp umožňuje instalaci OS MS Windows XP na počítači firmy Apple s procesorem Intel.

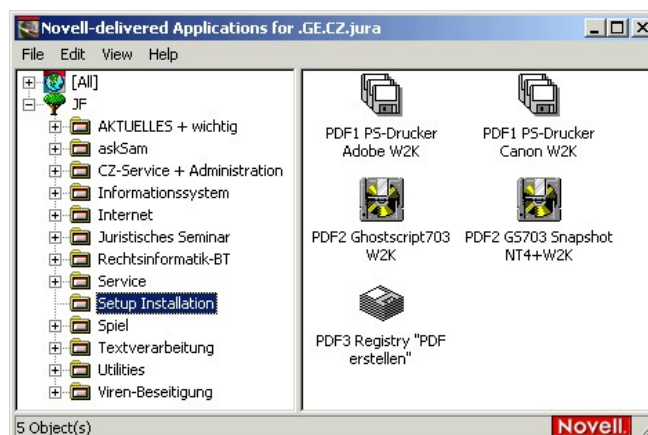
3.6 Multiplatformní

3.6.1 Novell - ZENworks

Novell.

Původně vzešlý z aplikace LANdesk Manager od Intellu, kterou si Novell licencoval. Dříve znám pod jménem NAL (Novell Application Launcher). Postupným přidáváním funkcionality přestal původní název odpovídat možnostem aplikace a systém

byl přejmenován na ZENworks (Zero Effort Networks). Systém se skládá z několika modulů: Desktop Management, Personality Migration, Software Packaging, Patch Management a další. Ve verzi 7. přichází s možností aplikovat uživatelské politiky na Linuxové stanice. Správa je centralizována do webového prohlížeče komunikujícím se SOAP službami. <<http://www.novell.com/products/zenworks/>>



Obr. 3: Novell – přehled SW vybavení

Obecné vlastnosti ZENworks				
Multiplatformní	I18N / čeština	Modulární	Konzole / GUI	Repozitář SW
ano ^{*1)}	ano / částečně	ano ^{*2)}	ne / ano	ne

1) podporovány jsou jen některé funkce

2) jedna varianta pro oblast

3.6.2 Postavené na jazyce Java

Aplikační servery postavené na J2EE – <<http://java.sun.com/javaee/5/docs/tutorial/doc/>>

Sun Java System Application Server, BEA Weblogic Server, IBM Websphere Application Server, Oracle Application Server, JBoss Application Server, JOnAS – jde o odlišnou koncepci správy SW, vycházející z možností jednotného prostředí JVM.

Hyperic HQ – Popdora (Linux, Windows, Solaris, HP-UX, Mac OS X.) Postaveno na technologiích J2EE, JBoss, Portable C, Ajax. Umožňuje používat pluginy v jazycích: XML, J2EE, Bash, C++, Perl, Ruby, Python, PHP, C#, apod.

<<http://www.hyperic.com/>>

SmartFrog – S distribucí instalačních balíčků v rozlehlých sítích. Podpora pro: HP-UX, Linux, Mac OS X, Windows <<http://www.smartfrog.org/>>

3.6.3 Postavené na jazyce Python

Bcfg2 – Plná podpora: FreeBSD, GNU/Linux (Debian (and Ubuntu)), Gentoo, distribuce založené na balíčcích RPM (CentOS, Mandrake, Redhat, RHEL, SLES, SuSE), Solaris.

Částečná podpora: AIX1, Mac OS X2

<<http://trac.mcs.anl.gov/projects/bcfg2/>>

Arusha – Podpora pro: Linux, Mac OS X, AIX, BSD, HP-UX, Solaris

<<http://ark.sourceforge.net/index.html>>

3.6.4 Postavené na jazyce C nebo C++

Cfengine – Podpora: Unix, Linux, Windows <<http://www.cfengine.com/>>

LCFG – Podpora pro: Redhat Linux. Částečná podpora: Linux, Solaris, Mac OS X

<<http://www.lcfg.org/>>

STAF – Není budován přímo jako systém pro správu SW. Jeho primární zaměření spočívá v testování aplikací na různých platformách. Obsahuje propracovaný centrální systém spouštění, řízení a kontroly aplikací na stanicích. Jde o open source projekt. Plná podpora: Windows 95, Windows 98 (a 98 SE), Windows Millenium Edition, Windows NT 4.0, Windows 2000, Windows XP, Windows Server 2003 (IA32, IA64, AMD64), Windows Vista (IA32, AMD64), Linux (Intel32/64, AMD64, PPC32/64), Linux on zSeries (31-bit, 64-bit), AIX Version 4.3.3.0 a <= (32-bit, 64-bit), Solaris (Sparc) 2.6 a <=, Solaris AMD Opteron 64-bit, Solaris (x86) 10 a <=, HP-UX 11.00 (PA-RISC, IA64 32-bit and 64-bit), OS/400 V5R2, z/OS UNIX V1.4, FreeBSD 4.10, 6.1.
<<http://staf.sourceforge.net/index.php>>

3.6.5 Virtuální prostředí

Jedním z modelů správy SW je i provozování aplikací ve virtuálních strojích, případně v terminálovém provozu. Terminálový provoz je podporován systémy několika firem. Kvalitní zázemí má, v prostředí Unixu a jeho klonů. Je to dáno jeho architekturou a historickým vývojem Unixu, který byl pro terminálový provoz připravován již na svém začátku.

Pro bližší seznámení s touto problematikou doporučuji věnovat pozornost projektu The Linux Terminal Server Project <<http://www.ltsp.org/>>.

3.7 Zhodnocení s ohledem na možnost využití v prostředí základních a středních škol

Správa software ve školním prostředí má několik charakteristických rysů.

- Velký poměr počtu uživatelů na jednu stanic. Průměrně 5 stanic na 100 žáků. Tento stav kopíruje metodický pokyn Ministerstva školství, mládeže a tělovýchovy „Standard ICT služeb ve škole“ [11] *.
- Vysoká migrace uživatelů, nezřídka i stanic.
- Zvýšené riziko nechtěného nebo záměrného destruktivního chování vůči systému.
- Poměrně velká skupina dětí mateřských škol a prvňáčků, kteří neumí číst a vyžadují přizpůsobené uživatelské prostředí a aplikace.
- Velký počet aplikací které mají uživatelé k dispozici. Především jde o desítky výukových programů a mnoho aplikací, se kterými se žáci učí pracovat.
- Žáci s tělesným či duševním handicapem, kteří mají specifické požadavky.
- Potřeba jednotného pracovního prostředí.
- Malá počítačová gramotnost ze strany učitelů. (Která se však postupně lepší.)
- Malé procento administrátorů odborníků. Není povědomí mezi řediteli, kde odborníky hledat a zajistit.
- Relativní nedostatek financí. Chaotická situace v přidělování dotací a financování dalšího rozvoje.
- Mnoho starší techniky, která je však svým výkonem dostačující pro téměř všechny výukové programy a velkou část, ve školství uplatnitelných, aplikací.

Mnohé, z komplexních řešení správy SW, jsou finančně mimo možnosti českých škol. Existují podpůrné granty, které tuto situaci pomáhají řešit. Většinou však školy nemají odborníky, kteří by takový grant byly schopni sestavit a obhájit. Snahy řešit tuto situaci celorepublikově, přináší přes obrovské náklady jen minimální účinek. Někdy jsou dokonce i překážkou rozvoje.

Školy mají možnost získávat repasovanou techniku, vyřazovanou z bank a dalších komerčních podniků, které pravidelně obměňují svou hardwarovou základnu. V současné době (2007) jde o počítače s OS Windows 2000 případně i Windows XP. Prostředí komerčních firem neklade na HW výkon počítačů takové nároky jako prostředí plné multimediálních programů. I z tohoto důvodu je vhodné provést na těchto stanicích

* Tento standard však vychází z chybného předpokladu využití počítačů převážně jen pro výuku Informatiky. Jednou z priorit, mého působení v roli hlavního administrátora několika základních škol, bylo docílit poměru 20 stanic na 100 žáků, čehož se navzdory aktivitám organizací, jako Správa informačních technologií města Plzně, úspěšně dařilo dosahovat.

downgrade na OS MS Windows 98 SE, který má řádově nižší nároky. Výrobci výukových programů, jelikož mají dobře zmapovanou HW situaci, jsou na současnou situaci přizpůsobeni. Většina výukových programů funguje na Windows 98 bezproblémově. Komerční komplexní řešení pro správu SW často již OS MS Windows 98 příliš nezohledňují.

Správa SW ve školství je, díky svým specifikám, pro mnoho firem nelukrativní, obávaná a systematicky ignorovaná oblast. Ze společností, které zde působí, většina nabízí produkty, které jsou určeny pro prostředí komerčních firem a jen obtížně se přizpůsobují potřebám školství. Z mála firem, které se na oblast správy SW ve školství zaměřují, většina prosazuje systémy, které fungují jen na majoritním typu OS. Mnohdy však nedisponují funkcemi a ovládáním, které by bylo, pro pověřené pracovníky na školách, vyhovující. Používání těchto programů není připraveno pro laickou veřejnost.

Některé potřebné funkce jsou opomíjeny. Nástroj pro třídění zástupců aplikací podle tematických charakteristik, nebyl součástí ani jednoho ze systémů správy SW.

3.8 Standardizační snahy v oblasti systému správy SW

V průběhu hledání dostupných řešení pro správu SW jsem zjistil, že se zde významně angažují některé velké standardizační organizace. Tato skutečnost výrazně zasáhla do záměrů mé práce.

3.8.1 Linux Standard Base

(LSB) – Organizace jejíž snahou je ustanovit standardy v prostředí Linuxu. Určuje metodiky ověřování podpory těchto standardů. Provádí certifikaci. Její aktivity vedou k usnadnění součinnosti mezi aplikacemi a operačním systémem. Snahou je, přiblížit Linux potřebám komerčních distributorů, výrobcům SW i uživatelům.

2.ledna 2007 bylo vydáno zhodnocení konference, která proběhla 4.12.2006 v Berlíně v prostorách SAPu. Na této konferenci se sešli přední představitelé balíčkovacích systémů Linuxu se zástupci nezávislých vývojářů software (ISV). Mezi závěry ke kterým dospěli patří i následující návrh. „*Best real-world solution was to construct a single API (application programming interface) that could be implemented across the various package systems.*”.

Na konferenci v Berlíně byla ustanovena nová pracovní skupina LSB Packaging workgroup <<http://www.linux-foundation.org/en/Packaging>>, jejíž cílem je iniciovat a podporovat snahy vedoucí k zjednodušení instalací a správy software třetích stran. Především jde o návrh jednotného API pro univerzální balíčkovací systém.

3.8.2 Solution Deployment Descriptor

(SDD) – zabývající se oblastmi nasazení SW a oblastmi životního cyklu aplikací. Funguje pod organizací OASIS. *„The purpose of this Technical Committee is to define XML schema to describe the characteristics of an installable unit (IU) of software that are relevant for core aspects of its deployment, configuration, and maintenance.”*

Jedním z výsledků jejich aktivit je dokument popisující XML schéma s informacemi o software, umožňující jeho instalaci, konfiguraci a údržbu

[<http://www.w3.org/Submission/2004/SUBM-InstallableUnit-DD-20040712/>](http://www.w3.org/Submission/2004/SUBM-InstallableUnit-DD-20040712/)

a dokument popisující XML schéma formátu univerzálního instalačního balíčku

[<http://www.w3.org/Submission/2004/SUBM-InstallableUnit-PF-20040712/>](http://www.w3.org/Submission/2004/SUBM-InstallableUnit-PF-20040712/).

3.8.3 Software-Based Communication - Domain Task Force

(SBC) – Skupina pod organizací OMG –The Object Management Group – *„Mission is the development of specifications supporting the development, deployment, operation and maintenance of software technology targeted for software defined communication devices.”*

[<http://sbc.omg.org/>](http://sbc.omg.org/)

3.8.4 JSR 88: Java™ EE Application Deployment

V rámci aktivit Java Specification Requests. *„This specification provides a complete description of the APIs required by the J2EE platform to enable development of platform-independent deployment tools.”* [<http://jcp.org/en/jsr/detail?id=88>](http://jcp.org/en/jsr/detail?id=88)

4 Návrh systému pro usnadnění správy SW

Při volbě tohoto úkolu jsem vycházel z dosavadních zkušeností a ze zprostředkovaných zkušeností kolegů administrátorů. Za dobu praxe administrátora jsem měl možnost setkat se se systémy správy SW různých rozsahů. Od nejmenších, pokrývajících jednotlivé stanice a servery, přes velmi malé a střední sítě s jednotkami až desítkami stanic. V roli certifikovaného správce informačních a komunikačních technologií jsem se podílel na správě zahrnující tisíce serverů, několik desítek tisíc stanic a několik set tisíc uživatelů.

Na všech úrovních jsem se setkával s řadou nedostatků a omezení systémů správy SW. Tím více jsem byl při studiu oblasti překvapen rozsahem dostupných teoretických poznatků i praktických výstupů.

4.1 Výchozí předpoklad

Neexistence komplexního, uživatelsky jednoduchého, multiplatformního a finančně nenáročného nástroje pro správu SW, který by trivializoval většinu rutinních administrátorských činností.

4.2 Zjištěná situace

Existence velkého množství komerčních řešení, která svými možnostmi přesahují požadavky na správu SW v malých a středních firmách. Existence open source i freeware řešení. Neexistence řešení, které by pokrývaly celou problematiku správy SW.

Obecně nízká znalost dostupnosti těchto řešení mezi administrátory. Sporadická existence vzdělávacích oborů v ČR, které by se této oblasti systematicky věnovaly.

4.3 Návrh obecných vlastností

Multiplatformnost – vývoj aplikace, je náročnější na návrh, realizaci i testování. Jak však ukazuje úspěch mnohých multiplatformních aplikací, jde o správný požadavek. Negativa snižuje dostupnost technologií jako je Java, Python a v jisté míře i .NET. U systému správy SW, který by měl být těsněji provázán s operačním systémem, je vhodné vytvořit vrstvu, která bude zprostředkovávat komunikaci s prostředím. V konečném důsledku může být negativum náročnějšího vývoje kompenzováno kvalitnějším návrhem a architekturou.

Modulární rozšiřitelnost – opět jsou zde za vzor brány úspěšné multiplatformní programy, Mozilla Firefox, Open Office, Eclipse, Gimp a další. Výhodou je upravitelnost systému na míru konkrétním potřebám. Předpřipravení aplikace pro funkce, které nejsou v době návrhu známy ani plánovány.

Internacionalizovatelnost – systém je směřován do prostředí, kde není dobrá znalost cizího jazyka samozřejmostí. Je tedy od začátku počítáno s připraveností pro překlad a pro přizpůsobení národním zvyklostem.

Textové i grafické prostředí – primárně je práce se systémem orientována na grafické uživatelského prostředí. Měl také obsahovat možnost plnohodnotné obsluhy pomocí textové konzole. Každé prostředí má své klady a zápory. Systém správy SW by neměl být omezen pouze na výhody jednoho přístupu.

Bezpečnost – se vzrůstající centralizací řízení do jednoho místa, je třeba věnovat větší pozornost i jeho zabezpečení. Nesprávně zvolená akce, nebo napadení systému, může ohrozit činnost mnoha stanic.

Hierarchická organizace systému – k zajištění škálovatelnosti i usnadnění spolupráce mezi všemi uživateli systému by mělo jít řešení, které by spojovalo klady centrálního řízení i decentralizovaného přístupu. Inspirací může být fungování DNS.

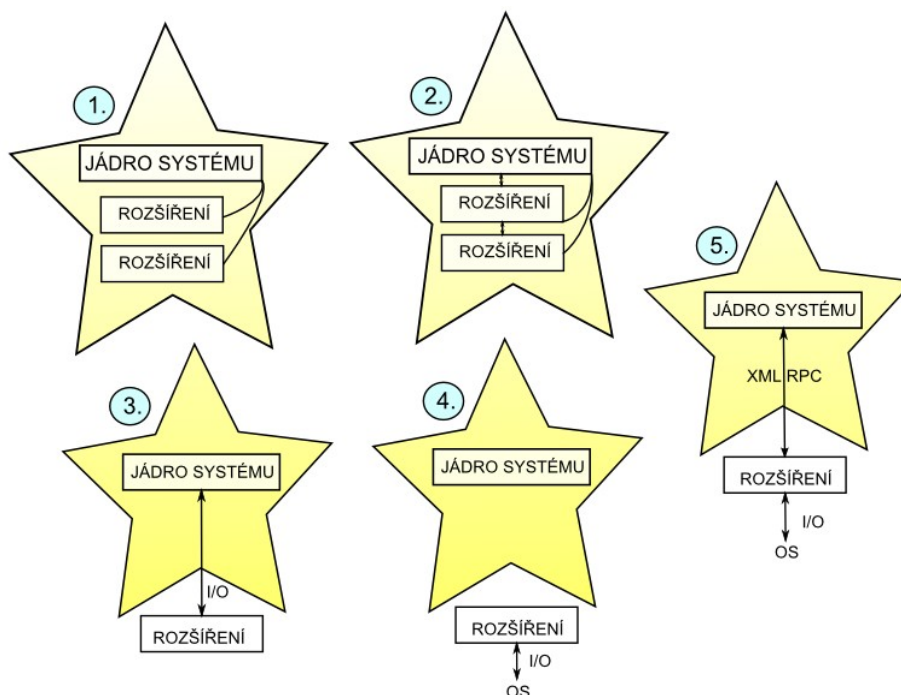
4.4 Úroveň jednotlivých stanic

Program by měl fungovat i na výkonově slabších strojích (CPU Pentium 75MHz, 32MB RAM, 1GB HDD), aniž by vyčerpал příliš mnoho systémových zdrojů. Hlavní OS platformou jsou OS MS Windows. Podpora je třeba zajistit pro MS Windows 98 SE a vyšší. Mezi další podporované OS by měly patřit Linux (Intel32/64, AMD64) od jádra 2.4, Solaris AMD Opteron 64-bit, Solaris (x86) 10 a vyšší, FreeBSD 4.10 a vyšší. Komunikace programu v sítích TCP/IP ipv4 a ipv6. V OS by měla řídicí část systému správy fungovat i jako daemon / služba, v privilegovaném i neprivilegovaném režimu. Dle konkrétních potřeb.

Spuštění a běh jednotlivých rozšíření může probíhat více způsoby. Liší se možnosti komunikace s jádrem aplikace a ostatními extensions i chováním vůči OS.

1. Jako modul, který je za běhu dynamicky připojen a zavolán jádrem. Vykoná svůj kód a předá řízení zpět jádru.
2. Jako modul, který je za běhu dynamicky připojen a jádro jeho kód spustí jako nové vlákno.
3. Jako externí program. Z pohledu OS jde o nově spuštěný program, který má ale přesměrovaný vstup, výstup a chybový výstup do jádra. Jádrem může být spuštěn v blokujícím režimu, čeká se na jeho ukončení, nebo paralelně s dalšími extensions.

-
4. Jako externí program. Z pohledu OS, standardní aplikace. Jádro na program nemá přímou vazbu.
 5. Jako externí program, kterému je při spuštění předána adresa a port xml-rpc serveru obsluhovaného jádrem. Z pohledu OS jde o samostatnou aplikaci, která po TCP/IP na daném portu komunikuje s jádrem. Defaultně s jádrem na localhost, s možností přesměrování na jinou IP a port.



Obr. 4: způsoby spuštění extension

Úkolem jádra je detekce prostředí ve kterém je spuštěno. Nalezení konfiguračních souborů jednotlivých rozšíření. Zpracování konfiguračních údajů. Sestavení plánu akcí. Vykonání akce podle plánu. V případě potřeby poskytuje rozšířením zjištěné informace o prostředí a je prostředníkem při komunikaci mezi rozšířeními. Samotné jádro nevykonává žádné akce vztahující se ke správě SW. Tyto funkce plní rozšíření. V příloze na straně 66. je ukázka jednoho z prvních návrhů pro tento systém.

Rozdělení funkčnosti na jednotlivé moduly by mělo vést k, přehlednějšímu návrhu a struktuře kódu, opakované využitelnosti kódu (reuse), snadnější údržbě kódu a ladění, usnadnění refactoringu, snadnějšímu rozdělení mezi více vývojářů. Více viz [12]

Zabezpečení může probíhat souběžně, několika způsoby.

Externí řešení. Jádro bude při spuštění hledat rozšíření na více místech v adresářové struktuře a běžné stanice nemusí mít k některým rozšířením přístup. Automaticky mohou být dostupná rozšíření, které bude mít uživatel ve svém domovském adresáři.

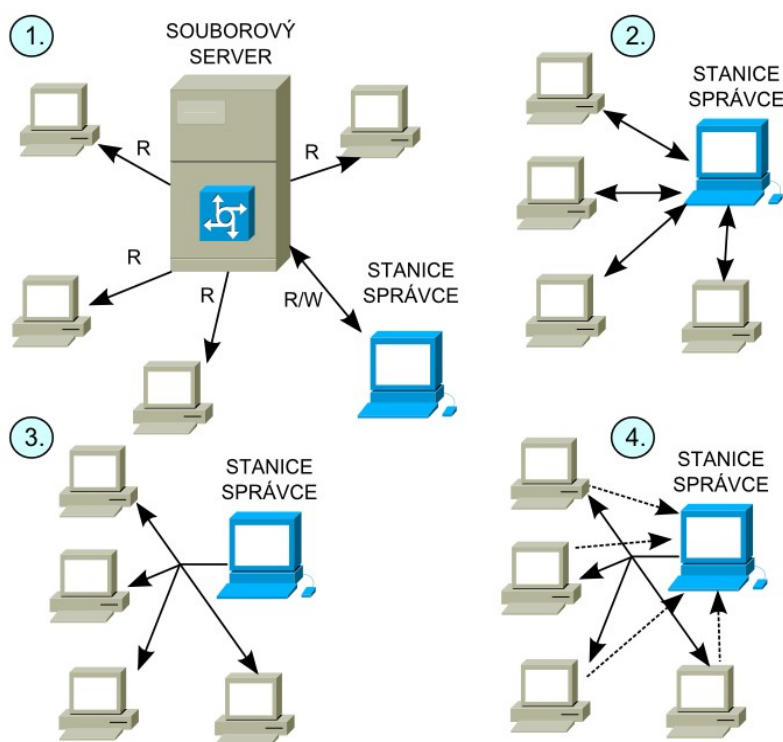
Řešení na úrovni jádra systému. Rozšíření budou dostupná, ale jejich aktivita bude vázána na zadání uživatelského jména a hesla, které bude ověřeno proti hash záznamu původního registrovaného hesla.

Na úrovni rozšíření. Každé rozšíření může být vybaveno vlastními metodami pro ověření a autorizaci uživatele. Přitom může ale nemusí spolupracovat s jádrem.

4.4.1 Úroveň lokální sítě

Při správě SW může být využíváno různých druhů scriptů. Login scriptů, scriptů vázaných na spuštění programu, scriptů spouštějících se v určitý čas. Všechny mají své nepopiratelné klady. Jsou však určeny k provedení konkrétní posloupnosti akcí a nejsou určeny k průběžně komunikaci se svým okolím.

Systém správy SW by se neměl omezovat na jeden způsob komunikace, ale měl by být přizpůsobitelný potřebám modulů. Několik způsobů komunikace se stanicemi viz Obr. 5.



Obr. 5: způsoby komunikace při řízení stanic

1. Tento způsob by se dal nazvat „broadcast souborem”. Stanice správce, která má právo zápisu do oblasti na souborovém serveru, sestaví a zapíše konfiguračního soubor. Klientské stanice se v pravidelných intervalech dívají, zda se nezměnil jeho čas uložení. V případě že zjistí změnu. Soubor si přečtou a zkusí v něm najít svůj

jedinečný identifikátor. Pokud svůj identifikátor najdou, provedou akci která následuje. (S tímto způsobem řízení mám praktické zkušenosti z předchozího realizovaného systému pro vzdálené hromadné řízení stanic.)

Výhody:

- veškerá síťová komunikace je transparentní, je obsloužena souborovým systémem a základním SW vybavením
- zabezpečení a přístupová práva jsou řešena transparentně
- komunikace může probíhat napříč celou sítí a není omezena kolizní doménou
- k zadání požadavků všem stačí řídicí stanici pouze jediná akce, zápis jednoho souboru na souborový server
- správce může provádět správu z kterékoliv stanice v síti
- konfigurační soubor je prostý text, lze používat běžné nástroje pro jeho úpravy, zálohy, kopírování, atd.
- pokud je souborový server dostupný i z Internetu, lze snadno provádět i vzdálenou správu stanic z textové konzole
- souborový server bývá spojen se switchem výkonnějším spojením, je proto vhodné komunikovat s jeho využitím
- řídicí soubor může být používán neustále i bez aktivní řídicí stanice
- identifikátor stanice v konfiguračním souboru nemusí být vztažen k IP adrese, lze používat DHCP bez pevných adres, lze definovat libovolné množství skupin na které se má akce aplikovat
- vhodné i pro přenosy velkého objemu dat, typicky image oblastí disku

Nevýhody:

- pokud je jediný řídicí soubor a více správců, kteří chtějí soubor používat, je možnost že si soubor přepíší, (lze řešit samostatným souborem pro každou stanici)
- v případě výpadku řídicí stanice ve chvíli kdy je konfigurační soubor naplněn nevhodnými úkoly, může docházet k nežádoucímu chování stanic, například akce vypnout počítač
- řídicí stanice nemá přehled, které stanice jsou aktivní a čtou konfigurační soubor, (zpětná vazba jde řešit podobným způsobem, zápis odpovědi na sdílený disk s právem zápisu)
- přepsání konfiguračního souboru novými údaji dřív, než jej stanice přečetly, (řeší nastavení timeoutu pro povolení k vložení další akce)

-
- negativem může být i despekt tohoto řešení ze strany sítířů
 - stanice musí stále kontrolovat změny v konfiguračním souboru, (lze optimalizovat dynamickým nastavením frekvence kontroly, defaultní hodnoty při provádění správy jsem nastavoval na 10 sec.)
 - při každé nové akci musí stanice parsovat konfigurační soubor, (otázka několika milisekund)
 - komunikace se souborovým systémem přes pomalé medium, (což příliš nevadí, protože souborový server používá cache)
 - ve srovnání s použitím socketů velká režie
 - čtení napůl zapsaného souboru, v praxi se tento jev neprojevil, konfigurační soubor je malý a z pohledu stanic se jeho zápis jeví atomicky, (lze ošetřit dodatečnými prostředky)
 - není vhodné pro složitou formu komunikace mezi řídicí stanicí a ostatními stanicemi kde záleží na milisekundách, například vzdálené ovládání GUI
2. Řídicí stanice vytváří samostatné spojení pro komunikaci se klientskými stanicemi. Mezi výhody tohoto způsobu patří přehled o aktivních stanicích, kontrola prováděných operací, rychlá síťová komunikace. V případě potřeby může řídicí stanice předávat data mezi stanicemi, které jsou v různých segmentech sítě. Podkladem pro navazování komunikace bývá statický seznam adres stanic.
 3. Vysílání řídicích zpráv broadcastem. Velmi praktické při předávání dat celé skupině najednou. Je bez kontroly doručení. Vysílání omezeno jen na kolizní oblast. Ta se dá rozšířit vhodnou stavbou síťových prvků, případně virtuálními sítěmi (VLAN)
 4. Kombinace broadcastu se zpětnou vazbou od klientských stanic. Vhodné k zjištění aktivních stanic a následnému navázání spojení. Případně je využíváno stanicemi jen k posílání zpětného potvrzení o přijetí nebo vykonání akce.

4.4.2 Úroveň organizace

Klíčovým prvkem této úrovně je dostatečně velký datový prostor a vhodný souborový server. Měl být schopen pojmout soubory s obrazy oblastí disků stanic. Rozdílové zálohy dokumentů a konfiguračních souborů z jednotlivých stanic. Součástí systému správy SW by měl být sklad instalačních balíčků aplikací, upravených (repackaging) pro potřeby nasazení v organizaci. Většina aplikací by však měla být použitelná bez instalace, spuštěním z připojeného sdíleného disku. Mobilní stanice by mohly mít vlastní mirror potřebných aplikací. V případě offline použití by se automaticky přemapovalo na lokální disk.

U organizací s rozsáhlejšími sítěmi může být zavedena hierarchická struktura rozvrstvení serverů, a potřebný obsah může být synchronizován s hlavním serverem. Pro zajištění vyšší dostupnosti a odolnosti proti poruchám, lze doporučit nasazení serverů ve virtuálním prostředí. Například XENu <<http://www.cl.cam.ac.uk/research/srg/netos/xen/>>.

4.4.3 Úroveň pro výměnu dat mezi organizacemi

Požadavky na správu SW vychází ze společných potřeb. Realizační postupy a prostředky mohou být také velmi podobné. V organizacích, které navrhovaný systém použijí, může vzniknout řada užitečných pomůcek, konfiguračních scriptů, nových rozšíření, vylepšení jádra systému. Podobně jako je tomu i jiných projektů (OpenOffice, Gimp, Firefox,), prostor věnující se sdílení doplňkových materiálů, může mít významný vliv na zkvalitnění a rozšíření mezi uživatele.

Vhodnou volitelnou funkcí je automatické zjišťování dostupnosti nové verze a nabídka její instalace. Samozřejmostí je připravení distribučních verzí a dokumentace. Společný prostor by měl disponovat diskuzním fórem, rubrikou FAQ, přehledem kam se projekt chce vyvíjet (roadmap), seznamem změn (changelog), aplikací pro informování o chybách (bugreport), nástroj pro sdílení kódu mezi vývojáři (SCM).

Pro open source projekty poskytuje server SourceForge <<http://sf.net>>, prostor pro všechny zmíněné požadavky. Existují i další s podobným zaměřením.



4.5 Modul pro usnadnění spouštění aplikací

Jako ukázkový modul pro systém správy SW jsem zvolil modul pro třídění zástupců a spouštění programů. Organizace zástupců je z pohledu komplexních řešení značně opomíjená oblast. Při návrhu grafických uživatelských prostředí se zřejmě vychází z předpokladu, že uživateli stačí pouze nevelké množství aplikací. Standardní funkcí správce oken (windows manager) je možnost umisťovat zástupce často používané programů na ploše. Méně frekventované programy mají své zástupce v rozbalovacím seznamu programů v hlavní liště.

Již při instalaci několika málo programů se začíná situace stávat nepřehlednou, ale při častějším používání si uživatel na umístění ikon zvykne. Co však v situaci, kdy je najednou k dispozici 200 programů? Ve školách, pro které je dostupnost velkého počtu aplikací charakteristická, se lze setkat se situací, kdy mají žáci k dispozici 500 i více programů.

Ukázka adresáře se zástupci programů ze základní školy viz příloha na straně 73.

Zástupce programů lze třídit do složek podle charakteristických vlastností. Mnoho programů však patří do více kategorií najednou. Je možné přistoupit k duplicitnímu umisťování zástupců do více složek, tím však stoupají nároky na jejich údržbu. I přes tuto

snahu nemusí být výsledek uspokojivý, protože předpokládá že si žáci názvy kategorií přečtou. Na základní škole s mateřskou školou a prvním stupněm může být téměř polovina všech žáků ve věku, kdy číst neumí. Situaci, kdy je způsob výběru zástupce a spuštění aplikace pro polovinu uživatelů velmi problematický, je třeba řešit.

4.5.1 V prostředí Linuxu

V prostředí textové konzole je běžné mít k dispozici najednou i několik tisíc programů. Běžné shelly jsou vybaveny funkcemi pro usnadnění zadávání příkazů. Podporují práci s programy ne jako se samostatnými jednotkami, ale jako se spolupracujícím a doplňujícím se systémem. Standardem je i přítomnost nápovědy k jednotlivým příkazům. Tento přístup nabízí mnohé výhody, klade však na uživatele specifické nároky. Je třeba, aby si uživatel alespoň přibližně pamatoval názvy příkazů, které chce používat.

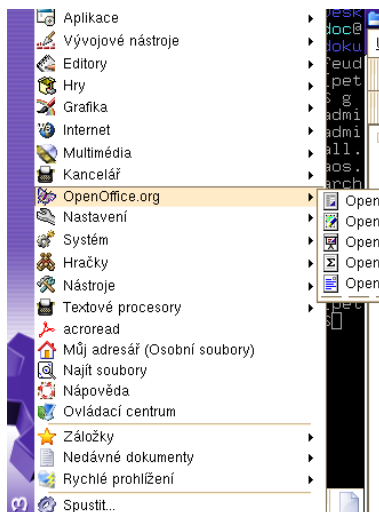
```
bszsplzen ~ #  
Display all 1981 possibilities? (y or n)
```

Obr. 6: Linux, příkazová řádka

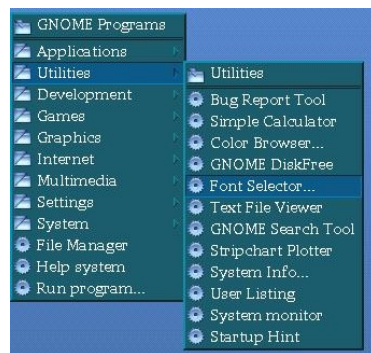
Grafická prostředí, kterých Linux nabízí celou řadu, disponují více virtuálními plochami a lepším řazením zástupců v nabídce programů. Součástí instalačního balíčku programu je i informace o tématické kategorii, do které má aplikace svého zástupce přidat.



Obr. 8: Linux, Elive



Obr. 7: Linux, KDE

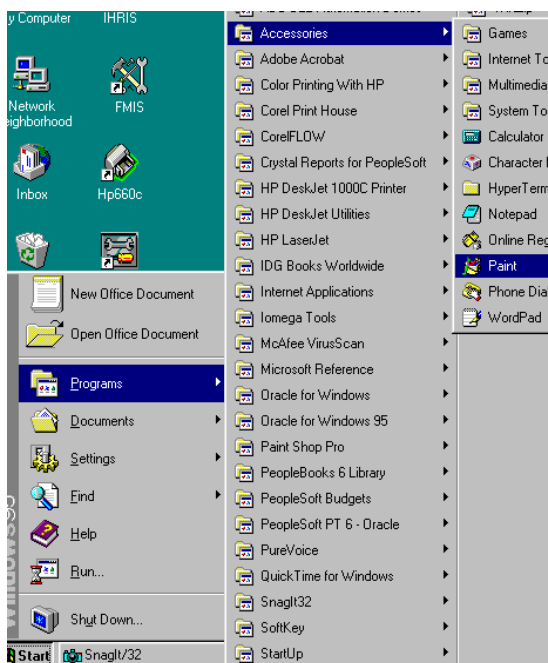


Obr. 9: Linux, fvwm

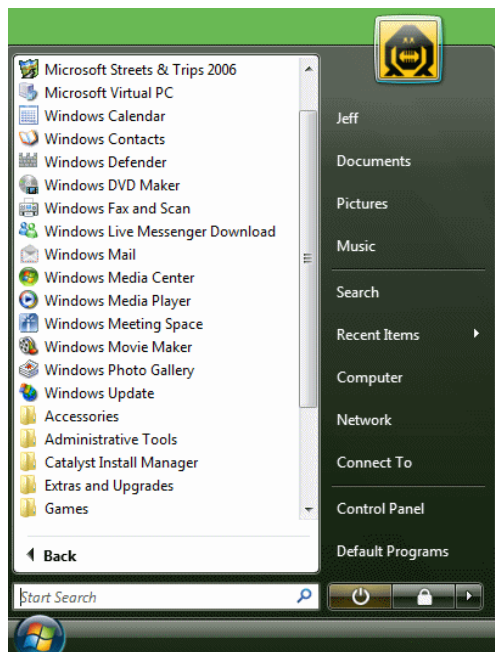
Se zájmem sleduji rozvoj v oblasti 3D grafických desktopů. I když většina funkcí těchto managerů nepřináší usnadnění, přesto se najednou výjimky. Už i jen to, že se lidé nad novými funkcemi zamýšlejí považují za dobré.

4.5.2 V prostředí MS Windows

V tomto prostředí je situace o poznání chaotičtější. Informace o typu aplikace není součástí instalace a je na libovůli aplikace, kam své zástupce umístí. Původní záměrem MS asi počítal s lepší organizací v zástupcích. Existují kategorie jako *Príslušenství*, *Hry*, *Systémové nástroje*. Aplikace však umísťují své zástupce po jména firem, pod názvy aplikací nebo přímo do hlavní nabídky. Situaci mírně pomohlo zavedením automatického skrývání zástupců málo používaných programů. Až ve Windows Vista došlo k výraznější změně. Panel start kombinuje funkci příkazové řádky a grafického menu. Do editačního pole lze napsat text, ke kterému jsou dynamicky zobrazovány související položky. Program se stejným chováním, Google desktop search <<http://desktop.google.com/>> je zdarma dostupný již několik let. V současné verzi podporuje pouze systémy Windows 2000/XP/Vista. S podporou starších verzí OS se nepočítá.



Obr. 11: Windows 98, nabídka



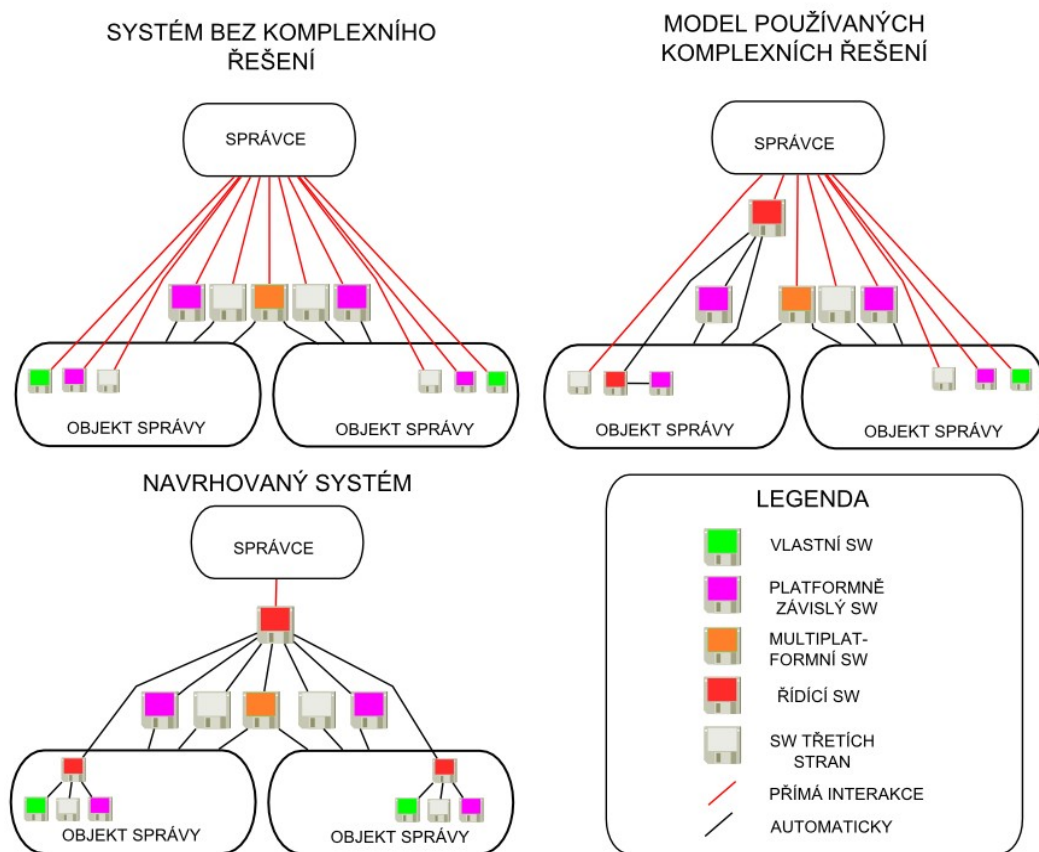
Obr. 10: Windows Vista, nabídka

Pro MS Windows existuje mnoho aplikací, které se uspořádání zástupců snaží řešit. Viz <<http://www.snapfiles.com/Freeware/system/fwtoolbars.html>>. Žádný z nalezených programů nepokrýval potřeby nasazení v základních školách. Některé z těchto požadavků jsou: Funkčnost i na Windows 98. Nenáročnost na systémové zdroje. Správa velkého množství ikon, 500 a více. Případná multiplatformnost. Snadná lokalizovatelnost. Třídění zástupců dle tématických skupin. Příslušnost zástupce ve více skupinách. Třídění najednou podle více skupin. Snadné doplňování skupin zástupců. Snadné třídění zástupců. Defaultní a personalizované nastavení. Volitelné zobrazení doplňujícího textu k zastupovanému programu. Možnost označovat si a doplňovat komentáře k zastupovanému programu. Volitelně, možnost spouštět programy kliknutím, místo dvojklíku.

4.6 Závěr návrhu systému pro správu SW

Cílem navrhovaného systému není vývoj nové aplikace, která by pokryla všechny oblasti správy SW. Cílem je vytvoření vrstvy, která by umožnila centralizovat a zautomatizovat komunikaci mezi správcem a aplikacemi. Pro integraci existující aplikace, by mělo stačit doplnit krátký popis. Jaké má závislosti. Jak se má spustit. Jak bude komunikovat. viz příloha, ukázka XML pro integraci rozšíření, na straně 68.

Ne všechny aplikace jsou vhodné pro tento způsob použití. V případě, že vhodná aplikace nebude dostupná, aby bylo možné použít vlastní rozšíření. Jádrem systému správy by mělo poskytovat služby, které rozšíření může, ale nemusí využít. Primárně by šlo o služby pro usnadnění integrace do prostředí a komunikace s ostatními aplikacemi.



Obr. 12: způsoby řízení správy SW

5 Implementace systému pro správu SW

Zdrojové kódy programu, programátorská dokumentace a videoukázky fungování aplikace jsou k dispozici na přiloženém CD. Aktuální verze jsou dostupné na adrese <<http://mintaka.eu/fav/kiv/dip2007/>> V zájmu podpory rozšiřitelnosti je dokumentace zdrojového kódu v anglickém jazyce.

5.1 Výběr programovacího jazyka a vývojových nástrojů

Výběr proběhl mezi programovacími jazyky C++, Python a Java. S ohledem na slabé hardwarové vybavení cílové skupiny, nebyla Java vybrána. Překážkou byly vyšší paměťové nároky a rychlost odezvy GUI. Jazyk C++ nebyl zvolen pro vyšší časovou náročnost vývoje. Jazyk Python byl zvolen s ohledem na to, že jde o interpretovaný jazyk. Obsahuje velké množství podpůrných knihoven, umí dobře spolupracovat s dynamickými knihovnami napsanými v C a C++, má podporu kvalitního vývojového prostředí Eclipse, podporuje několik grafických knihoven pro tvorbu GUI, aplikace je možné převést do binární podoby, osvědčil se jako multiplatformní jazyk pro systémové programování. V následující tabulce jsou uvedeny hlavní SW vývojové nástroje a prostředky.

http://www.python.org/	Programovací jazyk Python
http://www.eclipse.org/	Vývojové prostředí Eclipse
http://pydev.sourceforge.net/	PyDev – Podpora Pythonu v Eclipse
http://www.gtk.org/	GTK+ grafická knihovna pro tvorbu GUI
http://www.pygtk.org/	PyGTK – podpora GTK pro Python
http://glade.gnome.org/	Glade – návrh GUI
http://www.linuxjournal.com/article/7421	GladeGen – generátor kódu z Glade
http://effbot.org/zone/element-index.htm	ElementTree – XML parser pro Python
http://www.wmhelp.com/xmlpad3.htm	XMLPad – editace XML souborů

http://www.pspad.com/	PSpad – pomocný editor
http://www.vmware.com/	VMware – virtualizační nástroj
http://freemind.sourceforge.net/	FreeMind – průběžná externí dokumentace
http://epydoc.sourceforge.net/	EpiDoc – generátor dokumentace z kódu
http://www.gnome.org/projects/dia/	Dia – schémata a UML diagramy
http://dia2code.sourceforge.net/	Dia2Code – konverze z UML na kód
http://argouml.tigris.org/	ArgoUML – UML diagramy
http://www.gimp.org/	Gimp – Bitmapový grafický editor
http://www.inkscape.org/	Inkscape – Vektorový grafický editor

Tabulka 24: použité vývojové nástroje

5.2 Aplikace poznatků SW inženýrství při vývoji

5.2.1 Model-view-controller architektura

Použita na úrovni jádra aplikace i rozšíření. Pro implementované části jsou datovou vrstvou informace v XML souborech. Řídící část tato data připravuje a zpracovává. Uživatelskou interakci zprostředkovávají parametry příkazového řádku. U rozšíření, které zobrazuje zástupce, je to grafické prostředí generované knihovnou GTK. Architektura aplikace se snaží využívat modularitu ke zpřehlednění kódu a zmenšení oblasti změn.

5.2.2 Dokumentace

Auto-dokumentační řetězce v kódu Pythonu a jejich export do html, pdf, XML. Využíván nástroj EpiDoc. Dokumentace mimo zdrojový kód se odehrávala v nástroji FreeMind a její výstup lze najít na adrese <<http://mintaka.eu/fav/kiv/dip2007/>> ve formátu JavaApplet. K UML dokumentaci byly použity nástroje Dia a ArgoUML, ale častěji to byly tužka a papír.

Ukázka dokumentace zdrojového kódu viz přílohy strana 71.

5.2.3 Automatické testy

Python nabízí několik nástrojů pro automatické testování. Kromě self testů, unit testů, podpory pro automatické testování, GUI výstupy a vyhodnocování, obsahuje Python i nástroje pro testování GUI a specialitu, doctest, který podporuje tvorbu jednoduchých testů umístěných v dokumentačních komentářích.

<<http://docs.python.org/lib/module-doctest.html>>

Zpočátku bylo mým záměrem pokusit se vyvíjet tuto aplikaci technikou Testem řízeného vývoje. Bohužel se to pro počáteční stádium tvorby programu neosvědčilo. Aby testy mohly smysluplně fungovat, bylo třeba vytvářet mnoho mop objektů a příprava prostředí spolu s většími změnami v kódu nevedly k dobrým výsledkům. Tímto ovšem techniku automatických testů nezavrhuji a nadále ji považuji za užitečnou. O její užitečnosti svědčí i množství nástrojů pro podporu testování.

<<http://pycheesecake.org/wiki/PythonTestingToolsTaxonomy>>

5.2.4 Internacionalizace a lokalizace

V prostředí Pythonu je doporučován a často používán nástroj k usnadnění lokalizací Gettext <<http://docs.python.org/lib/module-gettext.html>>. Předpokládá se jeho použití. Podporu lokalizace má i použité grafické prostředí GTK, které ji řeší podobným způsobem. Prozatím implementovaná část pro komunikaci s uživatelem je určena i dětem, které neumí číst. Lokalizace byla prozatím odložena.

5.2.5 Logování

Omezení automatických testů bylo nahrazeno zvýšeným nasazením logování událostí. Mechanismus logování je podpořen interní knihovnou logging, která poskytuje několik úrovní logování, konfigurovatelnost výpisu a provázání mezi jednotlivými částmi aplikace. Na úrovni debug jsou logovány vstupy a vstupní parametry do metod, výstupy a výstupní parametry. Na úrovni info jsou logovány důležité interní akce, spuštění rozšíření apod. Na úrovni warning jsou výstupy nestandardních situací.

5.2.6 Správa verzí

Projekt byl zaregistrován a schválen na serveru SourceForge pod jménem LinkSorter <<http://sourceforge.net/projects/linkssorter/>>. V rámci SourceForge je k dispozici CVS úložiště. Zatím jsem však jediným vývojářem této aplikace a vzhledem k rychlosti odezvy sf.net bylo využito úložiště na lokálním počítači.

5.2.7 Refactoring

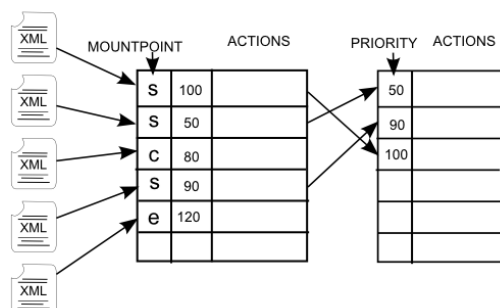
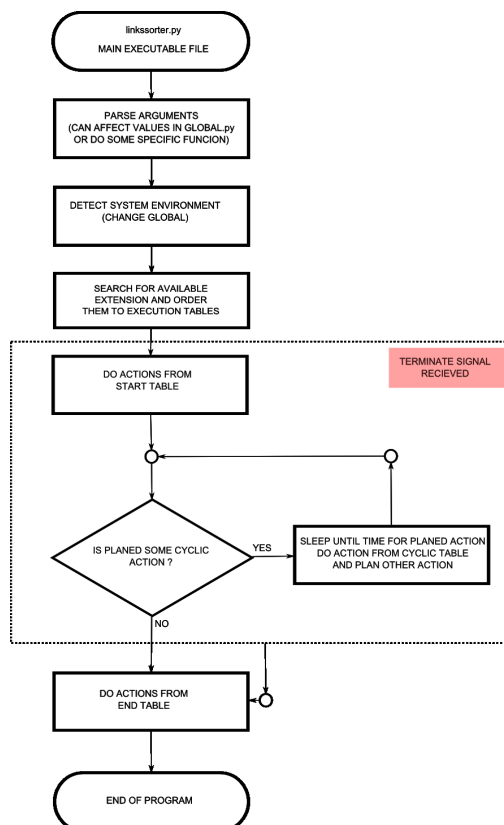
Refactoring byl využíván při drobných úpravách zdrojového kódu. Nejčastěji přejmenování názvů funkcí a proměnných. Výrazné použití této techniky bude prvním krokem v dalším vývojovém cyklu. K provádění refactoringu bylo využíváno služeb vývojového prostředí Eclipse s rozšířením PyDev.

5.3 Jádro systému

Úkolem jádra je zjistit, v jakém prostředí se program nachází. Nalézt konfigurační soubory. Získat data z konfiguračních souborů a sestavit z nich tabulky akcí. Provést naplánované akce v jednotlivých tabulkách.

Dostupné jsou tři tabulky: start, cyclic, end. Akce v tabulce start jsou provedeny při startu systému. V cyclic jsou akce které je třeba provádět opakovaně. Tabulka end slouží pro spuštění programů ve chvíli, kdy program dostal signál k ukončení nebo se sám dostal na konec.

Formát xml souboru pro popis rozšíření viz příloha na straně 68. Každé rozšíření může v tabulce akcí umístit libovolný počet přípojných bodů (blip). Každý nese údaj o tabulce, do které má být připojen (mountpoint), místu v tabulce do kterého má být zařazen (position), akci



kterou chce provést a způsob, jakým bude provedena. Do tabulek jsou vkládány podle priority celého rozšíření. Vyšší hodnota priority znamená, že rozšíření bude přidáváno později a bude mít možnost ovlivnit ta rozšíření, která měla nižší prioritu a byla přidána před ním. Po fázi přidávání jsou jednotlivé přípojky v tabulce seřazeny podle hodnoty parametru pozice.

Tímto je tabulka akcí připravena a je přidána do seznamu, který obsahuje odkazy na všechny tabulky akcí.

Poté jsou akce v tabulkách postupně vykonávány. Akce může být například:

- spuštění externího programu, který je na dané platformě spustitelný
- provedení akce se souborem, dle jeho typu určeného koncovkou, v závislosti na asociaci souboru v prostředí konkrétního uživatele.
- import modulu jazyka python, nebo dll knihovny a spuštění některé z jeho metod

Import a spuštění modulu může proběhnout jako spuštění nového vlákna, případně může být modul spuštěn v hlavním vláknu programu. V tom případě je běh jádra pozastaven, nejsou spouštěny další akce, dokud modul nevrátí řízení zpět.

5.3.1 Možnosti detekce HW a prostředí OS

Programovací jazyk Python nabízí k detekci prostředí několik modulů.

Funkce z modulu platform <<http://docs.python.org/lib/module-platform.html>>

Funkce	OS*	Popis
architecture	A	zjištění bitové architektury (32bit, 64bit) a formát spustitelných souborů
machine	A	název podporované architektury procesoru (i386, i486, i586, i686)
processor	A	jméno procesoru (Intel(R) Celeron(R) CPU 1.80GHz)
platform	A	identifikace prostředí (ve Windows XP "Windows-XP-5.1.2600-SP2") v Linux Gentoo (Linux-2.6.17-a-i686-Intel-R-_Celeron-R-_CPU_1.80GHz-with-glibc2.0 <)
release	A	identifikace vydání Linux (2.6.17-a), Windows XP (XP)
version	A	verze typu OS, číselná nebo časová identifikace
win32_ver	W	doplňující informace o platformě MS Windows získané z registrů ('XP', '5.1.2600', 'SP2', 'Uniprocessor Free')
mac_ver	M	doplňující informace o platformě Mac OS
dist	U	identifikátor distribuce
libc_ver	U	typ a verze standardní knihovny ('glibc', '2.0')
node	A	síťové jméno počítače (eryx.zcu.cz)
python_build	A	informace o sestavení aktivní verze Pythonu (1, 'Jun 13 2006 15:15:07')
python_compiler	A	čím byl zkompilován (GCC 3.2.3 20030422 (Gentoo Linux 1.4 3.2.3-r1, propolice))
python_version	A	verze Pythonu (2.4.2) major.minor.patchlevel
java_ver	A	informace o verzi JPythonu, pokud je v systému přítomen

Tabulka 25: funkce pro detekci prostředí

* A – obecná podpora, W – pouze pro MS Windows, U – pouze pro Unixové systémy, M – Mac OS

Další možnosti detekce prostředí se nachází ve specializovaných knihovnách. Jde například o funkce isdir (detekce adresáře) a obdobné isfile, islink, ismodule. Funkce pro čtení a vytváření systémových proměnných. Lze také parsovat výstup externích utilit jako lspci, netstat, nebo využívat prostředků konkrétního OS. U MS Windows lze řadu údajů zjistit z registrů pomocí funkce _winreg (Windows registry access) <<http://docs.python.org/lib/module--winreg.html>>, případně využívat knihovny pro parsování konfiguračních souborů .ini <<http://www.voidspace.org.uk/python/configobj.html>>. V prostředí OS Linuxu lze využívat ke zjištění údajů přístup do struktury /proc.

5.3.2 Možnosti úprav nastavení prostředí

Mnoho úprav prostředí je zabezpečeno přístupovými právy. K zajištění dočasného zvýšení práv programu, který byl spuštěn neprivilégovaným uživatelem, lze v Linuxu využít služeb `setuid`, případně `setgid` <<http://en.wikipedia.org/wiki/Setuid>> zavolané. V prostředí MS Windows lze využít systémové utility `runas`, která ale funguje na jiném principu. <<http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/runas.msp?mfr=true>>

V případě dostatečného přístupu lze přímo upravovat systémová nastavení v `/proc` nebo registrech. Lze měnit konfigurační soubory v `/etc` nebo `.ini` soubory. Lze využívat služeb jádra OS. Komunikovat s běžícími službami a daemony v OS. Je možné přímo využívat systémové knihovny.

Z aplikace Python lze ovládat i libovolné GUI aplikace. Pro prostředí MS Windows lze k tomuto účelu využít například knihovnu `PyWinAuto` <<http://www.openqa.org/pywinauto/>>, která se používá při automatickém testování GUI aplikací. Video–ukázku fungování tohoto programu lze vidět ve formátu flash, na výukovém serveru `ShowMeDo`. <<http://showmedo.com/videos/video?name=UsingpyWinAutoToControlAWindowsApplication&fromSeriesID=7>>

5.4 Rozšíření - vyhledání zástupců aplikací

V adresáři `extensions/linksminer/` je kód rozšíření pro vyhledání zástupců a skupin, kteří budou tříděni. Při své činnosti postupuje následovně:

- podívá se do adresářů, kde očekává konfigurační soubory s definovanými skupinami
- hodnoty zpracuje a vrátí v podobě odkazu na strukturu objektů
- podívá se do adresářů, kde očekává konfigurační soubory zástupců
- z nalezených údajů vytvoří strukturu objektů a vrátí ukazatel

Ukázku konfiguračních souborů skupin viz příloha na straně 69. Ukázku konfiguračních souborů skupin viz příloha na straně 70.

5.5 Rozšíření pro vzdálené spouštění aplikací

V adresáři `/extensions/runer/` je rozšíření, které umožňuje vzdálené hromadné spouštění aplikací. Porovnáním vlastní adresy počítače s hodnoty v konfiguračním souboru je určeno, zda bude vysílat požadavky nebo zda je bude vykonávat. Pro komunikaci je využito knihovny `socket`. Stanice, které poslouchají mají pro `socket server` vytvořeno další vlákno. V případě, že je učiněn požadavek na hromadné spuštění programu, vyšle řídicí stanice zprávu `broadcastem`. Obsah zprávy je tvořen řetězcem znaků. Cestou a jménem programu, který má být spuštěn. Stanice, která zprávu přijme, zkontroluje, zda je adresa odesílatele shodná s adresou řídicí stanice v konfiguračním souboru. V případě shody program spustí.

Toto rozšíření je použito spolu s rozšířením pro GUI třídění aplikací. Na řídicí stanici je k dispozici v menu Actions přepínací položka Remote Run. Zvolením této položky se přepne mód z lokálního spouštění na vzdálené. Stanice, které nejsou řídicí, mají tuto položku neaktivní.

5.6 Rozšíření - GUI pro třídění a spouštění aplikací

V adresáři /extensions/lsgtkpilot/ se nachází rozšíření, které používá datové struktury připravené rozšířením linksminer. Z údajů o skupinách sestavuje tlačítkovou lištu s přepínacími tlačítky. Z údajů o programech vytváří přehled ikon, které zobrazuje v hlavním okně. Umožňuje třídit zástupce podle kombinace zvolených skupin v tlačítkové liště. Zástupci mají svou ikonu a textový popis. U levého okraje hlavního okna je posuvná lišta, kterou se dá zvětšit prostor pro zobrazení popisu programu.

Grafické prostředí je postaveno na knihovně GTK s využitím aplikace pro návrh grafického prostředí Glade.

Ukázky GUI programu viz příloha na straně 74. Na přiloženém CD jsou k dispozici další materiály, včetně videoukázek práce s programem.

5.7 Nasazení v praxi

Tato aplikace je určena k náhradě současného systému pro správu SW, který je v současnosti používán na několika základních školách v ČR. Při jejím návrhu a tvorbě jsem vycházel z požadavků, které jsou na stávající systém kladeny. Aplikace byla testována na dvou školách v Plzni. V prostředí se systémy MS Windows 98 SE, 2000, XP. Průběžně je aplikace testována mezi žáky a učiteli Tyršovy základní školy v Černicích. Není nasazena v ostrém provozu, což je plánováno na začátek školního roku.

Před nasazením systému v praxi jsou plánovány dva vývojové cykly. První se zaměřením na výrazný refactoring a stabilizování současných funkcí. Následující cyklus se bude věnovat rozšíření funkčnosti, optimalizaci zátěže a lepší integraci do systému.

6 Závěr

V diplomové práci jsem se snažil přiblížit oblast správy software z pohledu člověka, který je k ní řadu let připoután. Množství let zřejmě nebude tím nejlepším měřítkem. Od začátku podrobnějšího zkoumání této oblasti jsem neustále nacházel další podstatné materiály a zdroje, o kterých jsem dosud nevěděl. Návrhy, postupy, standardizační doporučení a dokumentace, pomocné nástroje i komplexní aplikace. Mnoho z nových poznatků teprve bude prozkoumáno a utříděno. Ačkoliv jsem v této práci nemohl popsat všechny dostupné materiály, přesto doufám, že postupy a odkazy, zde uvedené, budou pro zájemce o oblast správy software užitečné.

I když nemohu slavnostně říci: „Zde je systém, který vyřeší všechny problémy správy software.“, jsem přesvědčen, že snaha zkoumat a řešit problémy této důležité oblasti je správná.

Přehled zkratek

API	Application programming interface, sada funkcí, které poskytují standardizovaný přístup ke zdrojům a službám, které aplikace s API nabízí
DHCP	Dynamic Host Configuration Protocol, protokol pro automatické přidělování IP adres a údajů potřebných ke komunikaci v síti
DNS	Domain Name System, systém pro distribuci informací určených k překladu zástupného jména na IP adresu
DOS	diskový operační systém, jeden ze starších operačních systémů
FAQ	často kladené otázky
GNU	General Public License
GTK+	multiplatformní knihovna pro tvorbu a ovládání GUI
GUI	Graphical User Interface, grafické prostředí pro komunikaci uživatele s počítačem
I18N	internationalization, techniky umožňující programu zohledňovat mezinárodní rozdíly při zobrazení znaků písma, data a času, měny a pod.
ID	identifikátor k jednoznačnému určení objektu vztaženého k této hodnotě
IP	Internet Protocol – standardní komunikační protokol pro výměnu dat
IT	oblast informačních a komunikačních technologií
JIT	installation hello
JVM	Java Virtual Machine – prostředí pro běh aplikací v jazyce Java
LSB	Linux Standard Base, projekt zaměřující se na standardizaci v prostředí Linuxu
MS	Microsoft, významná firma v oblasti IT
OASIS	Organization for the Advancement of Structured Information Standards, standardizační organizace
OS	Operating System – operační systém počítače, prostředník mezi HW a aplikacemi, základní program počítače, řídí a poskytuje jeho zdroje
RSS	Really Simple Syndication, protokol pro distribuci informací o změnách webových stránek založený na zasílání XML správ
SAP	významná firma v oblasti IT se zaměřením na podnikové informační systémy
SCM	Source Code Management, nástroj pro správu sdíleného obsahu
SOAP	Simple Object Access Protocol později znám jako Service Oriented Architecture Protocol, doporučený formát komunikace pro webové služby.
SSH	Secure Shell, protokol pro přenos šifrovaných dat, také program pro zabezpečený vzdálený přístup k počítači
TCO	Total Cost of Ownership, celkové náklady na vlastnictví
VLAN	virtual LAN, logická síť vytvořená za podpory aktivních síťových prvků
WFP	Windows File Protection, funkce pro ochranu souborů v OS MS Windows

Seznam tabulek

Tabulka 1: slovní označování verzí.....	12
Tabulka 2: výklady k autorskému zákonu.....	13
Tabulka 3: softwarové licence.....	13
Tabulka 4: servery zaměřující se na katalogizaci SW.....	14
Tabulka 5: odkazy k tematice BitTorrentu.....	15
Tabulka 6: managery souborů.....	16
Tabulka 7: utility pro správu souborů.....	16
Tabulka 8: SW pro evidenci aplikací.....	17
Tabulka 9: aplikace pro zálohu a obnovu diskových oblastí.....	17
Tabulka 10: nástroje pro záznam změn v systému.....	19
Tabulka 11: nástroje pro řízení kompilace.....	19
Tabulka 12: nástroje pro repackaging.....	20
Tabulka 13: virtuální mechaniky.....	20
Tabulka 14: programy pro vzdálenou správu.....	22
Tabulka 15: scriptovací nástroje.....	23
Tabulka 16: nástroje pro sledování a ovlivnění běhu aplikací.....	25
Tabulka 17: odinstalátory aplikací v MS Windows.....	28
Tabulka 18: pomocné nástroje pro tvorbu balíčků a konvertory.....	32
Tabulka 19: nástroje pro práci s formátem rpm.....	33
Tabulka 20: nástroje pro práci s formátem deb.....	33
Tabulka 21: zdroje k distribuci Gentoo a programu Portage.....	34
Tabulka 22: Nástroje pro interdistribuční instalaci aplikací.....	35
Tabulka 23: správa SW v OS BSD.....	35
Tabulka 24: použité vývojové nástroje.....	53
Tabulka 25: funkce pro detekci prostředí.....	56

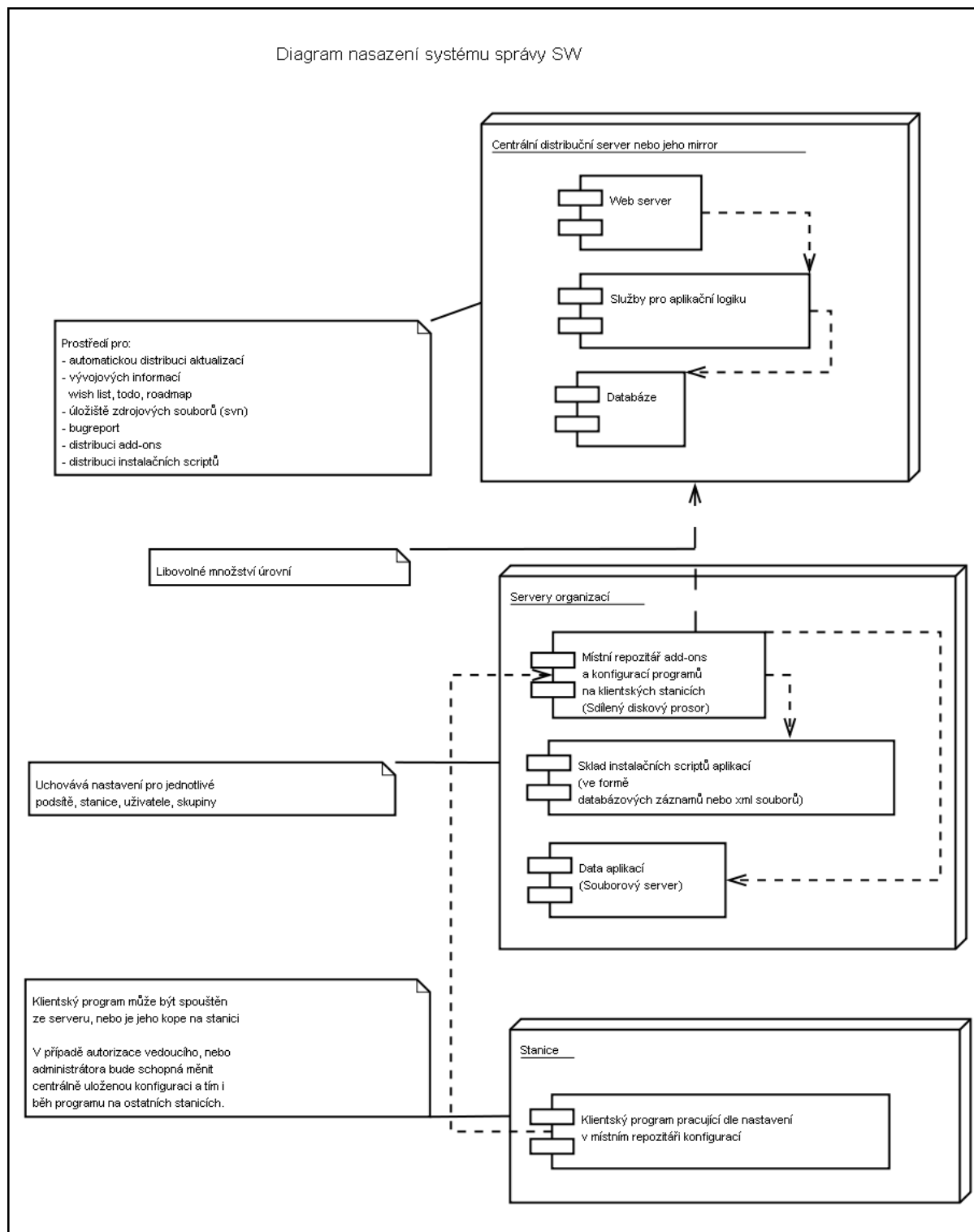
Literatura

- [1] *System Administrator History, 200x*, The Arusha Project [online][leden 2007]
URL<<http://ark.sourceforge.net/sysadmin-history.html>>
- [2] *IBM OS/360*, Wikipedie [online][2007]
URL<<http://en.wikipedia.org/wiki/OS/360>>
- [3] *Software versioning*, Wikipedie [online][leden 2007]
URL<http://en.wikipedia.org/wiki/Software_versioning>
- [4] *Pojem rootkit*, Wikipedie [online][únor 2007]
URL<<http://cs.wikipedia.org/wiki/Rootkit>>
- [5] Chris Long, *Windows Software Deployment*,
Realtime Publishers, [2003]
- [6] J. Ts, R. Eckstein, D. Collier-Brown, *Using Samba, 2nd Edition*,
O'Reilly & Associates, [2003]
- [7] *Použití sady nástrojů k vzdálené správě počítačů*, Microsoft [online][březen 2007]
URL<[http://support.microsoft.com/default.aspx?scid=kb;\[LN\];304718](http://support.microsoft.com/default.aspx?scid=kb;[LN];304718)>
- [8] Aleen Frisch, *Essential System Administration, 3rd Edition*,
O'Reilly & Associates, [2002]
- [9] Chris Long, *Software Packaging for Desktop Migrations*,
Realtime Publishers, [2003]
- [10] *Balíčky ve Slackware*, LinuxExpres [online][duben 2007]
URL<<http://www.linuxexpres.cz/praxe/balicky-ve-slackware>>
- [11] *Standard ICT služeb ve škole*, Ministerstvo školství, mládeže a tělovýchovy
[online][duben 2004] URL<<http://www.atre.cz/normy/page0213.htm>>
- [12] *Add-ons*, Mozilla developer center [online][únor 2007]
URL<<http://developer.mozilla.org/cs/docs/>>

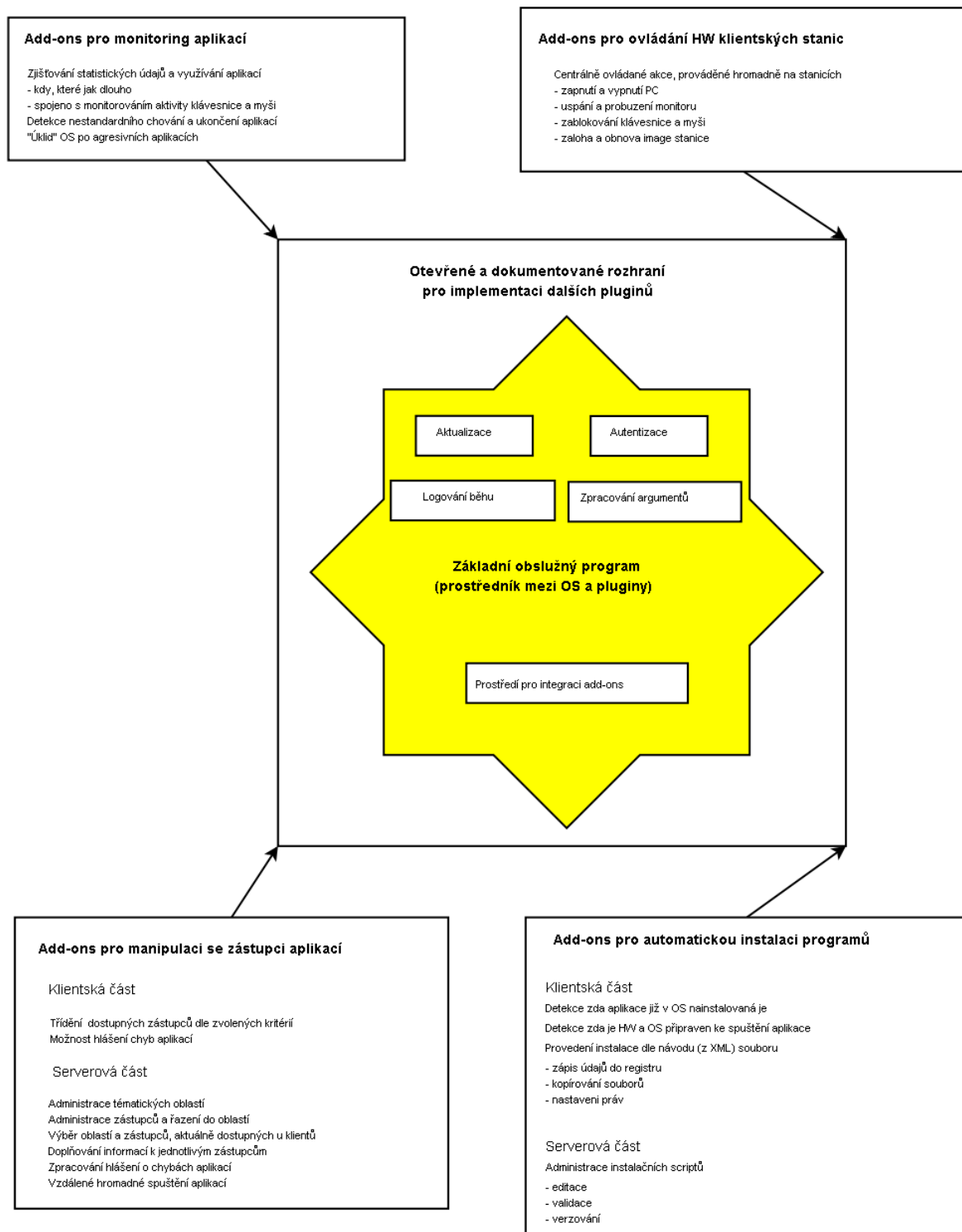
Přílohy

A) Diagram nasazení	65
B) Struktura řídicího programu.....	66
C) Rozšíření pro třídění a spouštění aplikací.....	67
D) Ukázka XML pro integraci rozšíření.....	68
E) Ukázka XML pro definování skupin.....	69
F) Ukázka XML s informacemi o zástupcích.....	70
G) Ukázka generované dokumentace.....	71
H) Logo projektu – Links Sorter.....	72
I) Ukázka současného způsobu třídění zástupců.....	73
J) Ukázky GUI programu pro třídění zástupců.....	74

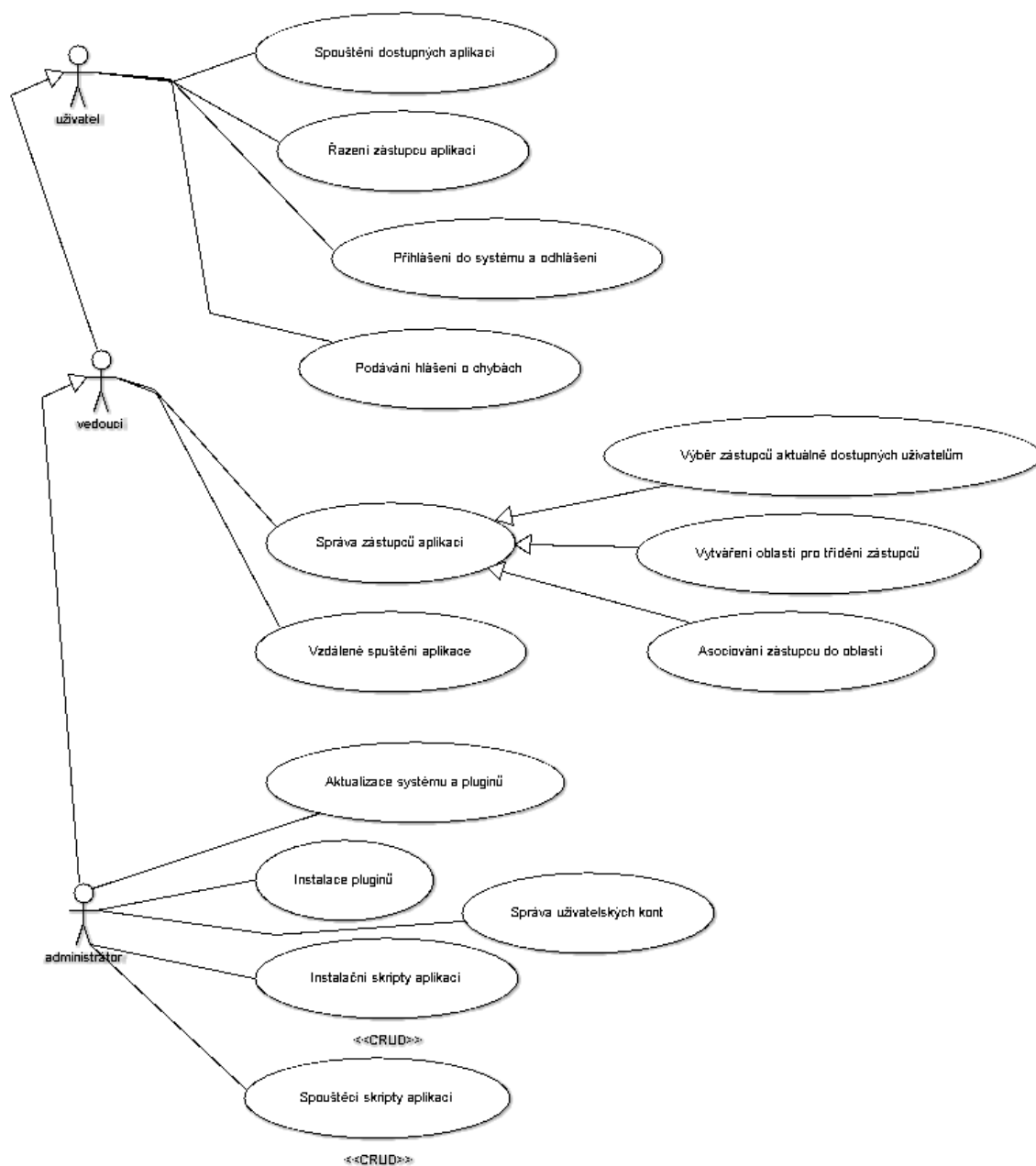
A) Diagram nasazení



B) Struktura řídicího programu



C) Rozšíření pro třídění a spouštění aplikací



D) Ukázka XML pro integraci rozšíření

```
<?xml version="1.0" encoding="utf-8"?>
<lsextend
  name      = "lsgtkpilot"
  version   = "0.1.2"
  priority  = "3000"
>
<!-- The lesser value means, that blip is write to executuio

  <description>Describe what the extension do</description>

  <blip> <!-- one or more blips describing activity of extens
    <mountpoint value="start"/>          <!-- in which par
    <position   value="3000"/>           <!-- label no. in
    <action     value="lsgtkpilot"/>     <!-- name of exec
    <parameters value="" />              <!-- parameters g
    <repeattime value="" />              <!-- only for cyc
    <runas      value="module"/>        <!-- method of ex
    <parallel   value="True" />         <!-- if blip run
  </blip>

  <dependency>                                <!-- various type
    <linkssort minversion = "0.0.1" maxversion = "0.9"/>
    <platform  name="" />
    <platform  name="" />
    <system    name="" />
    <extension name="" minversion = "" maxversion = "" />
    <extension name="" minversion = "" maxversion = "" />
  </dependency>

</lsextend>
```

|

E) Ukázka XML pro definování skupin

```
<?xml version="1.0" encoding="utf-8"?>
<lsgroups>
  <group>
    <name value="MŠ"/>
    <longname value="Mateřská škola"/>
    <iconpath value="./data"/>
    <icon value="./data/ms.xpm"/>
    <priority value="90"/> <!-- lesser value
    <separator value="False"/> <!-- If on toolbar
    <active value="False"/>
    <description>Text</description>
  </group>

  <group>
    <name value="1.st"/>
    <longname value="První stupeň"/>
    <iconpath value="./data"/>
    <icon value="./data/1st.xpm"/>
    <priority value="100"/> <!-- lesser value
    <separator value="False"/> <!-- If on toolbar
    <active value="True"/>
    <description>Text</description>
  </group>

  <group>
    <name value="2.st"/>
    <longname value="Druhý stupeň"/>
    <iconpath value="./data"/>
    <icon value="./data/2st.xpm"/>
    <priority value="110"/> <!-- lesser value
    <separator value="True"/> <!-- If on toolbar
    <active value="False"/>
    <description>Text</description>
  </group>
</lsgroups>
```

F) Ukázka XML s informacemi o zástupcích

```
<?xml version="1.0" encoding="utf-8"?>
<lslinks type="1">                                <!-- Type of lslinks.xml file structure -->
  <link>
    <id          value="07TerasoftAnglictinaProNejmensi001" />      <!-- [yy-C-P-V-L] yy y
    <version     value="070420"/>    <!-- Version of link -->
    <name        value="Angličtina pro nejmenší" />      <!-- Program name (required)-->
    <hash        value="" />        <!-- for security purpose // not implemented now-->
    <file        value="calc.exe" />    <!-- name of linked file (required) -->
    <filesize    value="" />        <!-- can be used for security // not implemented now-->
    <iconfile    value="ms.gif" />
    <iconfilesize value="" />
    <iconfilehash value="" />

    <absolutePat value="True"/>    <!-- if path is relative or absolute (optional) default T
    <path        value="" />        <!-- directory/path required) -->
    <arguments   value="" />        <!-- parameters given to script in action (optional) de
    <iconpath    value="./data/" />

    <startmessage value="" />      <!-- Show description before launching (optional) // not
    <startscript  value="" />      <!-- run some script before startig (optional) // not
    <endscript    value="" />      <!-- run some script after startig (optional) // not
    <installscript value="" />      <!-- run some script if program is not installed and dep

    <description>
      Program je vhodný pro děti od 7 do 12 let.
      Program je rozdělen do dvou hlavních částí - škola a hra.
    </description>

    <dependency>
      <platform    value="" />
      <system      value="" />
      <classcheck  value="" />      <!-- Script for checking class -->
      <class       value="" />
      <computerIP  value="" />      <!-- can be set xxx.xxx.* for group or xxx.xxx.xxx.[10,1
      <computerID  value="" />
    </dependency>

    <groups>
      <group name="" /> <!-- mean True if is in group False if group name is not here -->
```

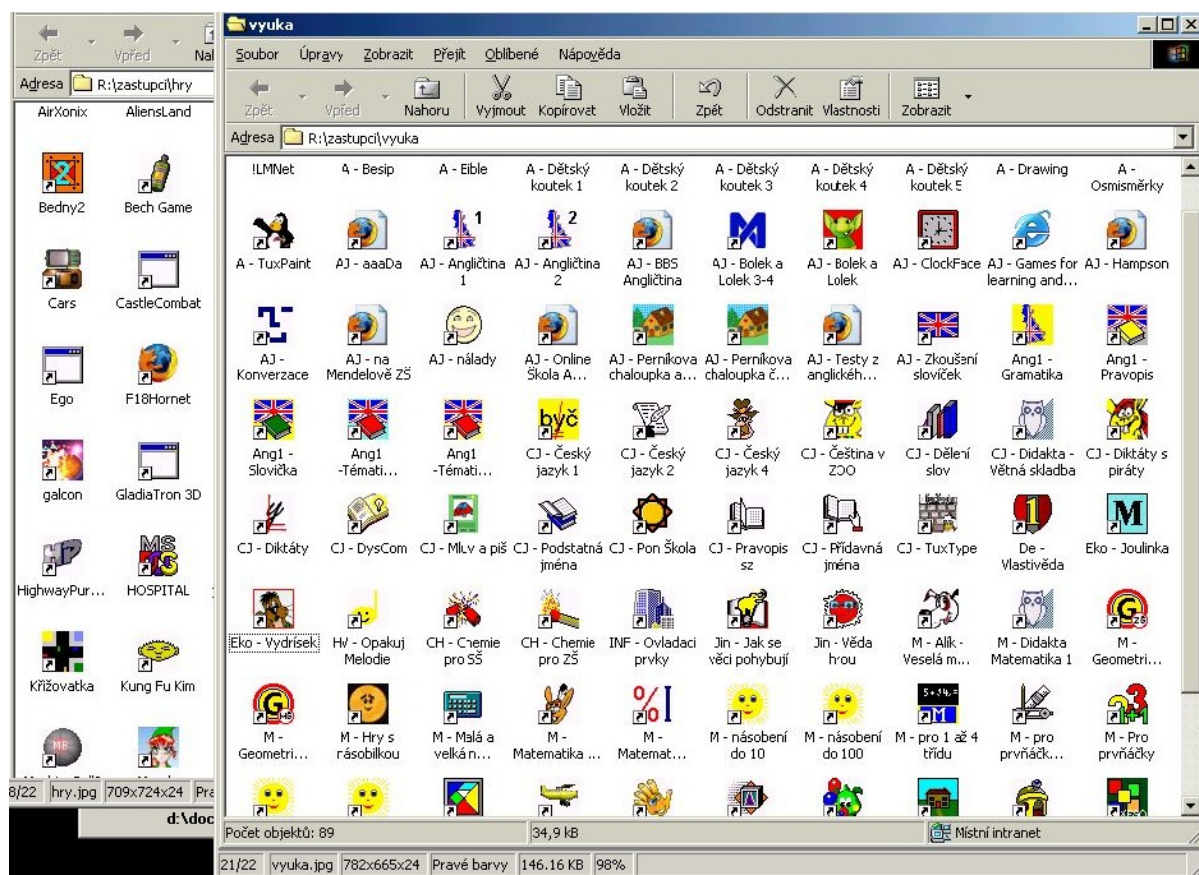
H) Logo projektu - Links Sorter

Projekt založený za účelem vývoje aplikace usnadňující výběr a spouštění aplikací.



<<http://sourceforge.net/projects/linkssorter>>

I) Ukázka současného způsobu třídění zástupců



J) Ukázky GUI programu pro třídění zástupců

Rozšíření, které slouží třídění zástupců a spouštění programů.

