

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

Bakalářská práce

Lokalizace produktů IT

Plzeň, 2009

Tomáš Dvořák

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 14.5.2009

Tomáš Dvořák

Abstract

Localization of IT products

This bachelor thesis deals with localization process and specific aspects of localization. Describes trends in localization, mainly on Google and Microsoft companies and explains some usual traps.

In second part is discussed concrete software product, design of its localization and described implementation of my localization process. Implementation part uses Java as a programming language.

Obsah

Prohlášení.....	2
Abstract.....	3
1.Úvod.....	5
2.Definice výrazů	6
3.Soudobé trendy.....	7
3.1 Google.....	9
3.2 Microsoft.....	12
3.3 Srovnání služeb Google a Microsoftu.....	13
3.4 Nástroje pro lokalizaci.....	13
4.Lokalizace.....	14
4.1 Co je třeba lokalizovat.....	15
4.2 Proces lokalizace.....	16
4.3 Konkrétní aspekty lokalizace.....	20
4.4 Problémy při lokalizaci.....	23
5.Návrh lokalizace.....	24
5.1 Zvolený software.....	25
5.2 Plán lokalizace.....	25
5.3 Detaily lokalizace.....	25
5.4 Lokalizace šablon.....	25
6.Implementace.....	25
6.1 Jaké možnosti nabízí programovací jazyk Java.....	26
6.2 O implementovaném softwaru.....	29
6.3 Detaily implementace.....	30
7.Závěr.....	36
Zdroje.....	37
Internetové zdroje.....	37
Tištěné zdroje.....	37
Přílohy	38
Obsah CD.....	38

1. Úvod

Tato bakalářská práce je zaměřena na navržení kompletního systému pro lokalizaci softwarového produktu, seznámení s jednotlivými aspekty lokalizace, problémy při lokalizaci a jejich řešení. Rozebírá standardní řešení a vysvětluje proč je navržené a implementované řešení pro konkrétní situaci lepší.

Lokalizaci produktu musí přecházet internacionalizace, tedy upravení produktu, především jeho zdrojových kódů, souborů s nápovědou a dokumentací do stavu, kdy je od sebe oddělen zdrojový kód a informace, které budou lokalizovány. Následuje lokalizace, kdy jsou zdrojové texty překládány do konkrétních jazyků, upravováno uživatelské nastavení, jako je směr písma, tvar výpisu data, měny, procent, adresy a mnoho dalších aspektů. To vše již bez zásahu do zdrojových kódů aplikace. Po lokalizaci nutně následuje testování a korekce k zajištění kvality finálního produktu.

V následujících řádkách bych chtěl poskytnout přehled o celém lokalizačním procesu. Také bych chtěl přiblížit jednotlivé principy fungování lokalizačních nástrojů a programů. Po přečtení této práce by měl mít každý dostatečný přehled o tom, co je lokalizace, jaké jsou postupy při lokalizaci a umět lokalizovat vlastní projekt.

2. Definice výrazů

Internacionalizace

Stručně řečeno, operace, po které jsou možné následné další lokalizace. Je to operace (jednorázová!), po které je program nebo množina programů začleněná do jednoho balíku schopna podporovat více jazyků. Je to zevšeobecnění, při kterém programy užívající pouze řetězce v jedné řeči, jsou spojeny do obecných cest, jak provést to samé v jiných jazycích.

Lokalizace

Lokalizace je operace, při které v již internacionalizovaných programech jsou poskytnuty informace takové, že mohou provádět své vstupy a výstupy ve tvaru, který je správný pro přirozené jazyky a kulturní zvyky. V podstatě internacionalizace je oblastí spíše programátorskou a lokalizace spíše překladatelskou.

Globalizace

Obecný pojem popisující proces návrhu, vývoje a adaptování produktu k distribuci ve více zemích. Možné také chápat jako kombinaci internacionalizace a lokalizace, ale je nutné vzít v úvahu i strategii a marketing pro vstup projektu na globální pole.

Překlad

Proces převodu textu do jiného jazyka. Je nutné zachovat původní smysl textu a přihlídnout ke kulturním odlišnostem.

Jazykové inženýrství

Odvětví zabývající se aplikací jazyka a textu v informačních systémech, tak aby mohly rozpoznat, pochopit, interpretovat a generovat lidskou řeč. Jazykové technologie zahrnují například automatický nebo počítačem předzpracovaný překlad, ověření mluvčího, získávání textu.

Locale

Sada parametrů, které definují jazyk, zemi a jiná nastavení, která chtějí uživatelé vidět v uživatelském rozhraní.

CAT – překlad podporovaný počítačem

Překlad, který je uskutečňován za podpory počítačového programu, navrženého k lokalizaci.

Překladová paměť (Translation memory)

Překladová paměť je databáze udržující segmenty, které již byly dříve přeloženy. Překladová paměť udržuje slova, fráze a odstavce a pomáhá překladateli při překladu nových textů. Paměť udržuje zdrojový text a korespondující překlad ve struktuře zvané jednotka překladu.

3. Soudobé trendy

Růst trhu

Lokalizace probíhá v poslední době do nových jazyků, do takových, které dříve nebyly považovány za atraktivní. Například malé jazyky, jako je slovenština, slovinština, nebo litevština. Pro velké firmy typu Microsoft nebyly dříve tyto jazyky nějak zajímavé, každopádně to se mění. Konkrétně Microsoft nemá ještě značné množství produktů lokalizováno ani do slovenštiny. Velká část těchto překladů je řešena strojovým, automatizovaným překladem.

Authoring

Authoring je psaní textu, například nápovědy nebo dokumentace, úplně od začátku. Na vytváření dokumentace se může podílet více lidí, a software pak zajistí, aby text vypadal jako od jediného autora. Sjednotí terminologii a styl.

Globalizace webu

Většina počítačových materiálů určených širší veřejnosti je produkována v angličtině. Ale již dnes je jen 30% uživatel internetu anglicky hovořících. Okolo roku 2015 by jich mělo být již jen 25 %. Proto je velmi důležité materiály lokalizovat. Následující tabulka nabízí představu o tom, jak je lokalizace důležitá pro firmy.

Firma	2003	2005	2006
BMW	16	32	34
Coca-Cola	16	26	26
Dell	14	21	25
eBay	10	9	12
Google	65	97	115
IBM	16	32	32
Palm	9	10	10
Starbucks	5	9	10
UPS	11	13	13

*Tabulka 1: Počet jazyků
podporovaných na webových
stránkách firem*

Preferované jazyky

V současné době se na webu velmi prosazuje čínština. Vzhledem k obrovskému tamnímu trhu je překlad produktů do čínštiny jednoznačně dobrá investice.

Lokalizací do 10 jazyků uvedených v tabulce 2. vlevo pokryjeme 80% všech návštěvníků webu. Tabulka 2. vpravo zobrazuje počty uživatel internetu hovořících daným jazykem.

Nejpopulárnější jazyky k lokalizaci	Jazyk uživatele webu	Celkem hovořících uživatelů
1. Angličtina	Angličtina	302,030,000
2. Němčina	Čínština	129,170,000
3. Francouzština	Španělština	83,710,000
4. Japonština	Japonština	80,000,000
5. Španělština (Španělsko)	Němčina	50,750,000
6. Čínština (zjednodušená)	Korejština	37,500,000
7. Italština	Francouzština	34,600,000
8. Španělština (latinská Amerika)	Italština	29,000,000
9. Korejšťina	Ruština	24,500,000
10. Portugalština (Brazílie)	Portugalština (Brazílie)	24,000,000

*Tabulka 2: Tabulka 10ti preferovaných jazyků k lokalizaci a
počet uživatel internetu, kteří hovoří daným jazykem*

Komunita a překlady

Některé země přejímají anglická slova velmi snadno, například Německo. U nás je to

s akceptováním angličtiny mnohem horší. Dobrou variantu získání správného překladu zvolil Google. O překladu slov jako „tab“ a „friend“ se rozhoduje na českých fórech a v diskuzích. O správných překladech tedy rozhodují přímo uživatelé, kteří budou software používat.

Ebay versus Aukro

Ebay je celosvětový aukční portál. České lokalizace se dočkal teprve na jaře roku 2009. Navíc nepřilíš povedené, kde není lokalizace ani registračních formulářů. Na jednu stranu je vidět snaha firem proniknout i na menší trhy jako je český, na druhou stranu s takovýmto přístupem se česká aukční jednička Aukro nemusí o svou pozici bát. Je vidět, že lidé dávají raději přednost místnímu produktu, který je celý v jejich rodném jazyce a nabízí českou podporu.

3.1 Google

Google je v současné době zřejmě největší hráč na webovém poli a své služby nabízí ve všech aktuálních oblastech zájmu. Lokalizace mu tedy nemohla uniknout. Nabízí několik služeb v základu odvozených od hlavního cíle Google, vyhledávání.

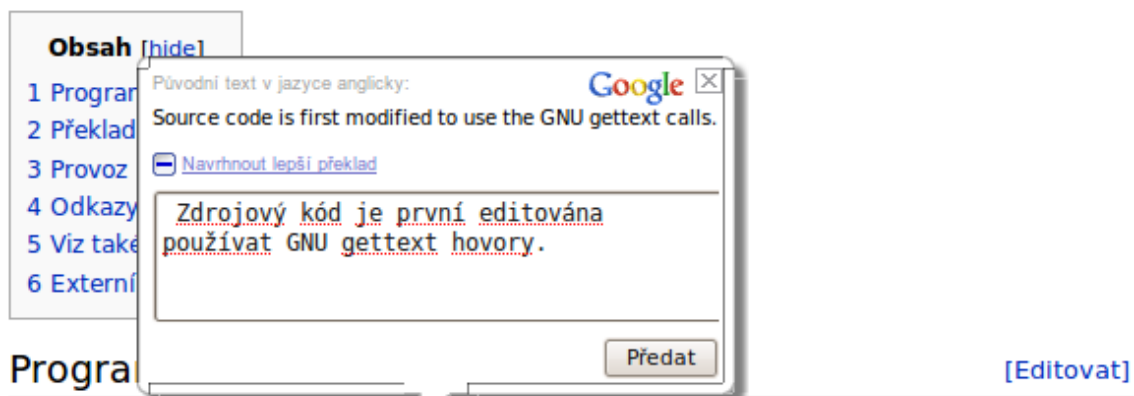
Slovník

Základním stavebním kamenem pro lokalizaci Google služeb je slovník, ze kterého čerpají ostatní aplikace. Slovník má i vlastní webové rozhraní, ale pro Google je to zřejmě jen doplňková služba, nad kterou je postaven mnohem složitější software.

Automatický překlad stránky

Pokud budete ve vyhledávači Google procházet výsledky vyhledávání a narazíte na stránku psanou jiným než výchozím jazykem, nabídne Google odkaz „přeložit“. Po kliknutí se zobrazí přeložená varianta stránky s možností zobrazit původní text a případně ručně editovat překlad, pokud máme zájem podílet se na zlepšení překladů.

Z ilustrace číslo 1 je vidět, že k čitelnému a srozumitelnému textu má překlad ještě daleko. Z části je to dáno technickým zaměřením textu ukázky, z části absencí ručně upravených překladů.

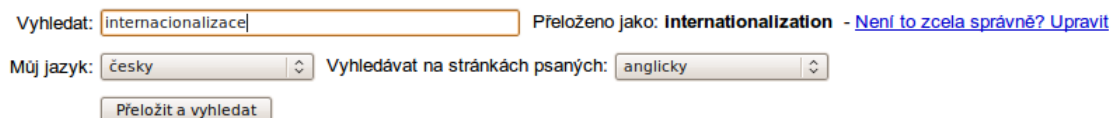


Zdrojový kód je první editována používat GNU gettext hovory. To je pro většinu programovacích jazyků, provádí balení strun, že uživatel uvidí v gettext funkce. Chcete-li ušetřit na psacím strojem času, a ke snížení kód nepořádek, tato funkce se běž

Ilustrace 1: Automatický překlad pomocí služby Google a možnost editovat text

Lokalizované vyhledávání

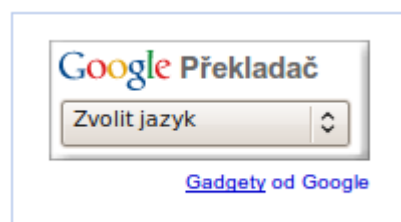
Google umožňuje zadat výraz v mnou preferovaném jazyce, zvolit cílový jazyk, ve kterém jsou napsány stránky a nabídnout výsledky hledání přeloženého výrazu a zároveň překlady těchto stránek (viz ilustrace 2.). Ve všech službách klade důraz na komunitu ochotnou spolupracovat a ručně editovat překlady a výsledky.



Ilustrace 2: Google lokalizované vyhledávání

API pro překlad stránek vytvořených uživateli

Pokud nechcete budovat vlastní řešení pro lokalizaci svého webu, nabízí Google jednoduchou miniaplikaci (ilustrace 3), která po vložení na stránky umožní uživateli změnit jazyk a prohlížet web přeložený Googlem tak, jak to dovede nejlépe. Včetně toho, že sami můžete upravit překlady a dosáhnout tak velmi dobrých výsledků.

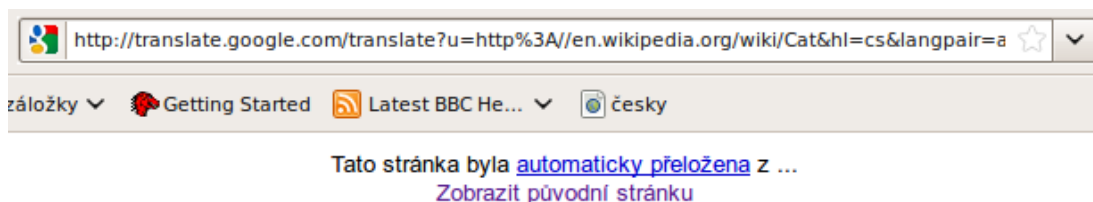


Náhled miniaplikace

Ilustrace 3: Google miniaplikace umožňující lokalizovat vlastní stránky

Integrace do prohlížečů

Google nabízí možnost umístit na panel Vašeho prohlížeče tlačítko (viz ilustrace 4.). Po kliknutí na něj bude aktuálně zobrazená stránka přeložena do zvoleného jazyka. Uspadňuje tak proces získání lokalizované varianty stránky.



Ilustrace 4: Google tlačítko pro lokalizaci aktuálně prohlížené stránky

Cílená reklama

Google adwords (ilustrace 5) je služba pro inzerci na portálu Google. Umožňuje cíleně oslovovat návštěvníky podle jejich polohy, klíčových slov nebo jazyka. Je to ideální způsob, jak propagovat svůj web přeložený do čínštiny jen čínským návštěvníkům.

在 Google 上为企业做广告

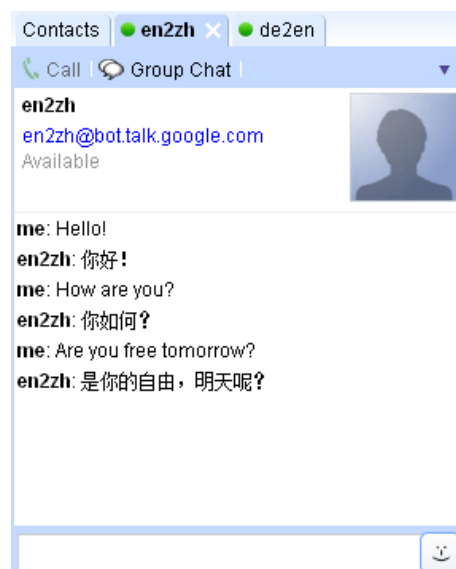
无论预算多少，您都可以在 Google 和我们的广告联网中展示广告。仅当有人点击您的广告时，您才需要支付费用。



Ilustrace 5: Google Adsense v čínštině

Google Talk bot

Google Talk je IM protokol postavený na bázi XMPP. V našich končinách je nejznámější IM zřejmě ICQ, Google Talk poskytuje podobnou funkčnost a je integrován do rozhraní pro poštu Gmail. Rozhovory vedené pomocí Talk je možné automaticky překládat pomocí Google Talk bota. Stačí k rozhovoru přizvat toho správného bota. Například pro překlad z angličtiny do čínštiny pozveme ke konverzaci bota en2zh@bot.talk.google.com. Ten se pak již sám o vše potřebné postará, jak je vidět na ilustraci 6.



*Ilustrace 6: Google Talk s
připojeným překladatelským botem*

3.2 Microsoft

Microsoft se specializuje na vývoj desktopových a serverových operačních systémů. Přesto však nabízí několik nástrojů s webovým rozhraním, použitelných k překládání textů.

Překlad textu

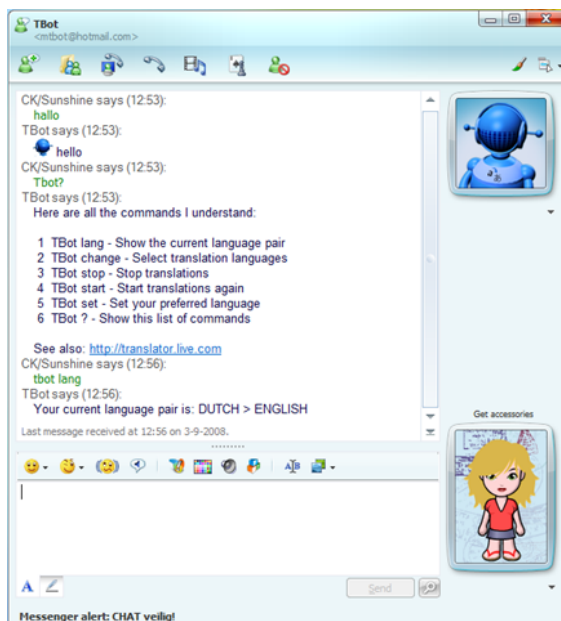
Podobně jako Google, umožňuje Microsoft překládat texty a slova z jednoho jazyka do druhého. Bohužel nabídka jazyků není zdaleka tak pestrá jako u Google, například čeština není dostupná vůbec.

Překlad stránek

Lokalizace stránek u Microsoftu probíhá odlišně než u Google. Není nabízena automaticky při vyhledávání na www.live.com, ale umožňuje zadat adresu, která má být přeložena. Výsledek se pak zobrazí v dvou-panelovém rozložení, kde nalevo je originální stránka a napravo lokalizovaná varianta.

Messenger bot

Velmi zajímavá funkce z dílny Microsoftu je Tbot. Tbot je kontakt v IM síti Microsoft messenger. Při jeho přidání do aktuálního rozhovoru bude automaticky překládat veškerou příchozí konverzaci. Umožňuje tak elegantní řešení, jak porozumět konverzaci s člověkem mluvícím nám naprosto neznámou řečí. Překlad je plně strojový, z toho plyne, že bude značně umělý a nepřesný, ale pro potřeby porozumění textu by měl být dostatečný. Ilustrace 7. ukazuje živý živý rozhovor překlad od T-Bota a možnosti jeho nastavení.



3.3 Srovnání služeb Google a Microsoftu

Nabídka služeb obou společností je vyrovnaná. Vyrovnaný už není počet jazyků, s kterými jsou aplikace schopné pracovat. Google nabízí 41 jazyků, Microsoft 13. Z části je rozdíl způsoben jiným zaměřením společností. Google se zabývá převážně internetovými aplikacemi, Microsoft aplikacemi pro stolní počítače.

3.4 Nástroje pro lokalizaci

V této kapitole bych se chtěl zaměřit na méně známé nástroje pro lokalizaci, poskytované v rámci open-source softwaru, tedy softwaru s otevřeným zdrojovým kódem. Tyto nástroje jsou použitelné legálně zdarma. Nástroj omegaT patří do kategorie CAT, kde kralují programy jako je SDL Trados nebo Déjà Vu. Jejich cena je ovšem v řádu stovek eur. Navíc je omegaT napsána v Javě. Proto jsem se rozhodl popsat ji.

Gettext

GNU Gettext je knihovna určená pro internacionalizaci a lokalizaci. Je běžně

používaná pro lokalizaci programů psaných v jazyce C a PHP. Použití je takovéto :

```
printf(gettext("Moje jmeno je %s.\n"), my_name);
```

Funkce gettext vrátí lokalizovaný řetězec, pokud takový existuje, případně původní variantu před překladem. Pokud touto funkcí obalíme veškeré řetězce v programu, nebude následná lokalizace žádný problém. Gettext umožňuje z celého adresáře se zdrojovými kódy vygenerovat soubory určené pro lokalizaci. Jeden lokalizační záznam pak vypadá například takto :

```
#: src/name.c:36
msgid "Moje jmeno je %s.\n"
msgstr ""
```

Je vidět, že k překladu je připojen i odkaz na soubor a řádek, ze kterého pochází. Je proto vhodné umístit do zdrojových souborů dokumentační komentáře, které pomohou překladatelům.

```
/// PREKLADATELE: Prosim nechte %s jak je, je potrebne samotnym sw.
printf(_("Moje jmeno je %s.\n"), my_name);
```

Takto napsané komentáře se pak zahrnou do lokalizačních souborů.

```
#. PREKLADATELE: Prosim nechte %s jak je, je potrebne samotnym sw.
#: src/name.c:36
msgid "Moje jmeno je %s.\n"
msgstr ""
```

Překladatel pak k položkám doplní překlad, například:

```
#: src/name.c:36
msgid "Moje jmeno je %s.\n"
msgstr "Je m'appelle %s.\n"
```

OmegaT

OmegaT je aplikace napsaná v Javě, pod licencí GNU, tedy svobodná a zadarmo. Aplikace poskytuje překladatelskou paměť. Slouží například k překladu dokumentů napsaných v OpenOffice. V původním dokumentu není třeba nic vyznačovat, jako je tomu u gettextu. Následně se pak text ve formátu například ODT (OpenOffice text) Otevře v omegaT, v levém sloupci bude právě zdrojový text, s vyznačením segmentace na jednotlivé bloky, v pravém sloupci překlady a nabízené překlady. Viz příloha 4.

4. Lokalizace

Jak již bylo vysvětleno v definici pojmů, lokalizace je přizpůsobení produktu místnímu

jazyku, zvyklostem a preferencím. Lokalizaci musí přecházet internacionalizace, které softwarový produkt dovede do podoby, kdy je lokalizovatelný. Jedná se především o oddělení zdrojových kódů od textů, automatické přizpůsobování šířky dialogů a nabídek, umožnění konfigurace klávesových zkratk atp.

4.1 Co je třeba lokalizovat

Lokalizace produktu se skládá v základě ze tří fází. Lokalizace samotného softwaru, nápovědy a dokumentace.

4.1.1 Software

Lokalizace softwaru je prvním krokem k lokalizaci celého projektu. Nápovědu a dokumentaci nelze lokalizovat dokud není lokalizován software, protože obsahují mnoho referencí na uživatelské rozhraní.

Objekty které je třeba lokalizovat :

Dialogová okna

Dialogová okna jsou okna nebo obrazovky, používaná ke změně voleb a nastavení. Většina systémů poskytuje okna se záložkami.

Nabídky

Nabídky se vysouvají z panelu programu nebo zobrazují po kliknutí pravým tlačítkem nad objektem(kontextové nabídky). Nabízejí spuštění příkazů, zobrazení dialogových oken a práci s programem. Položky označené třemi tečkami otevírají další dialogová okna a nastavení, položky bez teček okamžitě vykonávají příkaz.

Řetězce

Řetězce zahrnují chybové hlášky, otázky, bublinovou nápovědu atp. Ve většině aplikací je překlad těchto řetězců nejnáročnější úkol. Především proto, že řetězce nebývají kontextově závislé. Nabídky a dialogová okna většinou bývají seskupena a není problém prohlédnout si je pohromadě. Při překladu řetězců často překladatel jen odhaduje, kde a jak bude řetězec zobrazen.

4.1.2 Náповěda

Lokalizace nápovědy začíná jakmile je lokalizován samotný software. Lokalizace nápovědy je největší část práce. V době, kdy je téměř každý počítač připojen k internetu, se nápověda přemístila taky tam. Tím se proces lokalizace nápovědy transformuje na lokalizaci webu. Některý software obsahuje kontextovou nápovědu, je tedy důležité, aby obsahovala odkazy. Současné nápovědy obsahují množství multimediálních prvků.

4.1.3 Dokumentace

Ve většině případů dodavatel software k produktu přikládá online dokumentaci. Nejen, že se ušetří za tištění materiálů, ale snáz se zapracovávají aktualizace a opravy chyb. K software je pak přiložen jen jeden nebo dva tištěné dokumenty, většinou návod k instalaci a první kroky. Návod k instalaci provede uživatele instalací, první kroky pak ukážou základy používání softwaru. Detailní dokumentace je pak dostupná online. Další tištěné materiály přikládání k softwaru mohou být registrační karty nebo marketingové materiály, například upoutávky na další software daného výrobce. Jako formát online dokumentace se osvědčil Adobe Portable Document Format (PDF). Hlavní rozdíl mezi nápovědou a dokumentací je v použití, nápověda je často kontextově závislá, dostupná z různých částí aplikace, připravená například pro tisk vybrané části nebo problému. Dokumentace je ucelený dokument o softwaru.

4.2 Proces lokalizace

Každý projekt lokalizace by měl projít několika základními fázemi lokalizace, které zajistí, že nebude vynechán nějaký důležitý krok, který značně zkomplikuje práci překladatelům nebo i tvůrcům softwaru v konečné fázi projektu. Základní fáze jsou tyto:

1. Prezentační fáze
2. Počáteční sezení
3. Analýza zdrojových materiálů
4. Plánování a rozpočet
5. Stanovení terminologie
6. Příprava zdrojových materiálů

7. Překlad softwaru
8. Překlad online nápovědy a dokumentace
9. Testování softwaru
10. Snímky obrazovek
11. Návrh a tisk dokumentace
12. Zapracování změn
13. Zajištění kvality a doručení
14. Ukončení projektu

Prezentační fáze

V první fázi lokalizace projektu se obvykle vezme CD obsahující veškeré momentálně dostupné materiály k lokalizaci a rozešle se více firmám zabývajícím se lokalizací. Ty mohou na základě poskytnutých dat vyhodnotit náročnost a cenu překladu a podat nabídku.

Počáteční sezení

Jakmile je produkt představen, koná se sezení, kde se setkají manažeři, vývojáři a překladatelé. Probírá se zaměření produktu, úskalí, plány při lokalizaci.

Analýza zdrojových materiálů

K analýze materiálů k překladu se většinou poskytuje několik různých typů dat, je to především :

- Běžící verze produktu v jazyce, v jakém je vyvíjen
- Prostředí pro znovuvytvoření běžícího programu včetně zdrojových kódů
- Prostředí pro vytvoření nápovědy, včetně zdrojových kódů
- Složka s dokumentací, obsahující veškeré texty a grafiku

Pokud jsou dostupné, tak předchozí verze programu s překlady, z důvodu zachování terminologie.

Plánování a rozpočet

Naplánování je kritická část lokalizace. Většina softwarových firem vyvíjí software průběžně a průběžně ho dodává překladatelům. Tím se možnosti překladu posouvají ke konci projektu, kdy je projekt téměř dokončen a překladatelé jsou zahrnuti prací.

Stanovení terminologie

Většina projektů začíná vytvořením multijazyčného slovníku, vázícího se k dané aplikaci. Ten by měl být schválen vydavatelem projektu dříve, než dojde k jakékoli lokalizaci. Základní list terminologie, zvaný glosář, obsahuje běžně používané výrazy v uživatelském rozhraní, nápovědě a dokumentaci. Glosář by měl také jasně říkat, které fráze, jako například název projektu, by lokalizovány být neměly.

Příprava zdrojových materiálů

Poté, co inženýři a DTP odborníci projdou zdrojové materiály, stanoví problémové oblasti a proběhne zkušební kompilace softwaru, bývají obvykle vytvořeny překladatelské balíky. Příprava balíků obnáší například zjištění, zda neexistuje předchozí verze překladů, z které by se dalo čerpat. Pokud existují, použijí se. Tím se dosáhne větší konzistence překladů a zároveň se sníží počet nových slov.

Překlad softwaru

V ideálním případě je software první lokalizována věc. Překladatelé editují čistě textové soubory s klíči a překlady. Používají k tomu textové editory nebo přímo nástroje ze skupiny CAT. Doporučený postup překladu je nejprve přeložit dialogová okna, nabídky a nakonec řetězce. V okamžiku, kdy jsou přeloženy dialogová okna a nabídky, je možné vytvořit glosář pojmů. Většina referencí z nápovědy a dokumentace se odkazuje právě na nabídky a okna. Proto v tomto okamžiku můžeme vygenerovat testovací verzi programu a začít s překladem dokumentace a nápovědy. Do glosáře není dobré zahrnout všechny řetězce, stává se pak nepřehledný a méně použitelný.

Po vygenerování testovací verze programu probíhají testy správnosti překladů. Pokud je vše v pořádku, glosář se „zmrazí“. Poté je možné získat snímky obrazovek pro nápovědu a dokumentaci, poslat testovací verzi zákazníkovi ke schválení a pracovat

na překladech nápovědy a dokumentace.

Překlad online nápovědy a dokumentace

V okamžiku, kdy máme vytvořen glosář a testovací verzi programu, mohou práce na překladech nápovědy a dokumentace začít.

Testování softwaru

Testování softwaru předchází úprava zdrojových kódů, změna velikosti dialogů, tak aby se vešly i lokalizované řetězce, přiřazení unikátních klávesových zkratk a zkompilování programu do běžící verze.

Prvním testem by měla být jazyková kontrola, provedená překladatelem za pomoci lokalizačního experta. Často jdou použít skripty pro testování, které provedou testera celým procesem. Pokud skripty neexistují, záleží na dohodě vydavatele a společnosti poskytující překlad, kolik času a práce má být věnováno testování.

Snímky obrazovek

Snímky obrazovek jsou fotografie uživatelského rozhraní. Ukazují činnosti, které software umožňuje, dialogy s nastaveními a typické postupy práce. Snímky obrazovek se vytvářejí pro každý cílový jazyk. Bez snímků není možné dokončit online nápovědu ani dokumentaci. Proto je doporučeno začít snímky vytvářet, jakmile je dokončeno jazykové testování a terminologie je zmrazena.

Návrh a tisk dokumentace

V okamžiku, kdy jsou snímky obrazovek a veškerá grafika lokalizovány, je možné vytvořit návrhy dokumentace a nápovědy. Práce na dokumentaci a nápovědě mohou začít dříve, s nelokalizovanými obrázky, a poté je vyměnit za finální, lokalizované. To s sebou však nese problémy, jako je změna velikosti obrázku při změně velikosti dialogového okna v cílovém překladu. Tím je změněn tok dokumentu a budou třeba následné úpravy finální podoby dokumentů.

Zpracování změn

Překlad často začíná, zatím co se software stále vyvíjí. To znamená, že než je vypuštěna finální verze, je nutné průběžně zpracovávat změny i v lokalizaci. Pokud poskytovatel lokalizace nepoužívá překladovou paměť, je nutné kopírovat a vkládat opakovaně text. Používání překladové paměti umožňuje znovu použít texty automaticky. Pokud již začaly práce na dokumentaci a nápovědě nebo testování dokumentace, je zpracování změn ještě těžší.

Zajištění kvality a doručení

Před odesláním by měly být otestovány veškeré materiály a zajištěna kontrola kvality. Kontrola by měla zahrnovat minimálně tyto kroky :

- Korekce všech překladů
- Test přeložení softwaru a jeho odevzdání
- Dokončení oprav chyb a finalizace
- Ověření, zda byly splněny všechny body stanovené v úvodním ujednání o lokalizaci.

4.3 Konkrétní aspekty lokalizace

Při lokalizaci si každý asi představí překlad textu do daného jazyka. Situace je však mnohem složitější, a aspektů lokalizace je mnohem více, než jen čistý text. Pod lokalizaci spadají takové věci, jako grafické prvky, výpisy dat a měn, směr textu (LTR nebo RTL) a klávesové zkratky.

4.3.1 Grafika

Již při návrhu grafiky je třeba myslet na lokalizaci. V ideálním případě by měla být grafika navržena tak, aby samotný obrázek byl na pozadí a jednotlivé řetězce ve vrstvě nad obrázkem. Pak je možné text nechat upravit překladatelem a následně vygenerovat výsledný Jpg nebo Gif. Pokud je text již sloučen s grafikou, je velmi obtížné, ne-li nemožné, řešit jakoukoli lokalizaci.

Ikony

Při návrhu grafických prvků je důležité myslet i na kulturní odlišnosti států, pro které bude lokalizace probíhat. Nejlepší je vyvarovat se jakémukoli z těchto motivů:

- Symboly složené z písmen, během překladu se význam změní
- Grafika obsahující text
- Grafika zobrazující části těla a řeč těla
- Grafika zobrazující humor a slang
- Grafika zobrazující etnické, rasové politické nebo náboženské motivy
- Grafika která rozlišuje pohlaví
- Grafika zobrazující zvířata
- Grafika zobrazující sexuální a násilné motivy
- Grafika zobrazující místní konvence, jako je směr čtení a datum/čas

Některé CAD/CAM systémy mohou obsahovat až 5000 ikon a motivů, vznikla proto potřeba tyto symboly standardizovat. Tímto tématem se zabývá norma ISO/IEC 11581. Grafiku rozděluje na šest částí:

Část 1: Ikony - Obecné

Část 2: Ikony objektů

Část 3: ukazatele

Část 4: Ovládací prvky

Část 5: Ikony nástrojů

Část 6: Ikony akcí

Použití vlajek v lokalizaci

V aplikacích se často používá vlajka jako symbol jazyka. Jakou vlajkou ale reprezentovat angličtinu? Americkou, britskou, kanadskou nebo australskou? Všechny

tyto země mluví anglicky.

Vlajky jsou národnostní, reprezentují názory, hranice a politickou příslušnost, ale rozhodně ne jazyk. Proto by nikdy neměly být používány jako symbol jazyka.



Ilustrace 8: Vlajky jednotlivých států

4.3.2 Datum

Datum bývá zapisováno krátkým nebo dlouhým formátem. Krátký má tvar xx/yy/zz, kde kterákoli dvojice může znamenat den, měsíc nebo rok. Vše závisí na místních zvyklostech. Dlouhý typ data, například „April 22, 2009“ je již nezaměnitelný.

4.3.3 Měna

Při formátování měny je třeba brát ohled na tyto aspekty: symbol pro měnu, například Kč nebo €, často se využívá zkratka názvu měny, například CZK a pozice tohoto symbolu, tedy zda bude před nebo za číslem.

4.3.4 Číslo

U čísel jsou podstatné oddělovače tisíců a desetinných míst. Druhá problematická oblast je znázorňování záporných čísel. Záporná hodnota se značí například znaménkem mínus před nebo za číslem, nebo závorkami okolo čísla.

4.3.5 Adresa

Neexistuje standard pro adresu, ani pro směrovací číslo. Nemůžeme tedy například napsat jednotný test na správnost zadaného směrovacího čísla. U nás má směrovací číslo jen numerické znaky, blok po třech a blok po dvou oddělený mezerou. Ale například v Kanadě jsou to tři dvojice jakýchkoli znaků.

4.3.6 Webové adresy

IDN(internationalized domain name) je způsob, jak využívat v doménovém jméně znaky platné pro danou oblast. V České republice například www.háčkyčárky.cz. Problémy nastávají, pokud budete takovou adresu chtít zadat na klávesnici bez háčků a čárek. Další problém může být podvrh znaků, kdy například některé znaky z azbuky vypadají stejně jako latinka. Není pak problém napsat adresu, která vypadá stejně, ale vede úplně jinam. Celou touto problematikou se zabývá právě web www.háčkyčárky.cz

4.3.7 Kódování znaků

Nejvhodnějším kódováním pro lokalizaci je UTF-8, vyjmenujme několik jeho výhod:

Obsah znaků – UTF-8 obsahuje přes 2 miliardy jednotlivých znaků. Proto je možné v něm ukládat znaky různých národních sad

Zpětná kompatibilita – pro ASCII znaky je symbol v UTF stejný jako původní, není problém používat je

Celistvost jednotlivých znaků – první bajt znaku je odlišný od ostatních, je proto možné i při poškozeném znaku správně rozpoznat další znak. Toto není ani tak výhoda pro lokalizaci, jako pro programátora

4.4 Problémy při lokalizaci

Délka textu

Délka lokalizovaného textu hraje velkou roli při rozvržení grafického uživatelského rozhraní. Proto při lokalizaci do více jazyků bývá jeden z prvních testovaných německý. Při překladu z angličtiny do němčiny je nárůst délky textů v řádu desítek procent. Tomuto nárůstu délky musí být přizpůsobeny nabídky a dialogy, aby se vešla do okna aplikace.

Emulace zařízení

Častým problémem při lokalizaci do jazyků jako je hebrejštiny nebo arabština. Pro práci s takovým jazykem je potřeba mít zařízení s podporou tohoto jazyka. Při použití jiného

zařízení nebo emulátoru nebudou výsledky dobré, protože zařízení nemusí mít všechny fonty nebo plnou podporu daného jazyka. Nejlepší varianta je pořídit si zařízení určené přímo pro daný jazyk a testovat na něm, nebo alespoň využít kompletní lokalizační balík pro náš systém.

Barvy

Jednotlivé barvy nabývají různý význam v různých státech. Například v západních civilizacích je smuteční barva černá, v Číně a Japonsku je to bílá. Červená barva u nás většinou vyjadřuje agresivitu, v Číně a Korei je to barva štěstí a prosperity. V Indii je pro prosperitu vyhrazena žlutá.

5. Návrh lokalizace

V druhé části práce jsem se rozhodl lokalizovat svůj vlastní software Záložky (ilustrace 9.). Detailnější popis k softwaru je uveden v kapitole „Zvolený software“. K lokalizaci použiji stejný jazyk ve kterém byl vytvořen samotný program, tedy jazyk Java. Jeho možnosti jsou popsány v kapitole „Jaké možnosti nabízí Java“. V implementační části jsou postupně rozebrány jednotlivé aspekty lokalizace, vysvětlen jejich smysl a důvod, proč jsou implementovány jinak než standardně.



Ilustrace 9: Základní obrazovka aplikace Záložky

5.1 Zvolený software

Jako software, který je lokalizován jsem zvolil mou vlastní aplikaci „Záložky“, původně vytvořenou jako semestrální práce pro předmět KIV/DB1.

Záložky umožňují sdílet mezi uživateli zajímavé webové adresy, komentáře a soubory. Typické využití, pro které byl software vyvíjen je například sdílení informací a plánů o dovolené, kde několik uživatelů přidává do společného projektu odkazy na zajímavá místa, poplatky na dálnicích, informace o cílové zemi a jiné.

5.2 Plán lokalizace

Je potřeba vytvořit webové rozhraní pro administraci překladů, které by umožňovalo snadnou správu jazyků a překladů pro jazyky. Překlady by měly být drženy v databázi, aby k nim byl snadný přístup a snadno se udržovaly. Oproti jiným aplikacím bych se chtěl zaměřit na dynamiku přidávání jazyků za běhu systému.

5.3 Detaily lokalizace

Vzhledem k nápadům, které jsem chtěl do práce zahrnout, nebylo možné použít standardní lokalizační postupy. Vytvořil jsem tedy vlastní implementaci, postavenou na možnostech jazyka Java. Práce původně využívala JSP stránky pro prezentační vrstvu, ale protože udržování a úpravy těchto stránek jsou dost nešikovné, jako prezentační vrstva byl zvolen Velocity šablonovací systém. Ten značně zjednodušuje i postup lokalizace.

5.4 Lokalizace šablon

Pro implementaci jsem zvolil šablonovací systém Velocity, Lokalizaci šablon jsem implementoval sám, vzhledem ke specifickým požadavkům na překlady. Využito bylo Velocity macro, způsob jak použít část opakujícího se kódu ve všech zdrojových souborech. Makro umí podle daného klíče a locale vypsát lokalizovanou variantu.

6. Implementace

V části o implementaci softwaru bych se rád zmínil jaké standardní prostředky nabízí

jazyk Java, jako základní jazyk, ze kterého jsem vycházel, a poté popsal, které části jsem implementoval sám a jinak.

6.1 Jaké možnosti nabízí programovací jazyk Java

Jazyk Java je multiplatformní a multijazyčný. Nabízí standardní způsoby, jak lokalizovat zdrojové soubory. Lokalizace v Javě se odvíjí od aktuálně použitého locale.

Locale

Locale je objekt reprezentující specifické geografické, politické nebo kulturní odlišnosti. Operace, která potřebuje locale ke své funkci se nazývá locale-sensitive a používá locale jako zdroj informací pro svou práci, například zobrazení data, které se pro různé locale bude zobrazovat různě. Locale se skládá z několika identifikátorů. První je země, jedná se o dvouznakový kód podle normy ISO-639. Například pro češtinu je kód „cs“. Druhý identifikátor je země podle normy ISO-3166, pro Českou republiku je kód „CZ“.

Locale je potřebné pro formátování měny, procent, data a dalších údajů. Pro formátování je možné použít například třídy DateFormat nebo NumberFormat. Výstup těchto metod pro všechny dostupné locales je v příloze 1.

ResourceBundle

Resource bundle obsahuje objekty specifické pro dané locale. Kdykoli program potřebuje zdroj závislý na locale, může ho nahrát z Bundlu. Používání Bundlu umožňuje psát programy, které jsou maximálně nezávislé na locale. Resource Bundly sdílejí stejné jméno, ale rozlišují se přidáním identifikátorem, který rozlišuje locale. Například pro balík překladů MyResources bude mít německá varianta tvar MyResources_de a anglická MyResources_en. Každý Bundle obsahuje stejné položky, jen v daném jazyce.

Bundle pro konkrétní locale se získá následovně :

```
ResourceBundle myResources =  
    ResourceBundle.getBundle("MyResources", currentLocale);
```

ResourceBundle obsahuje dvojice klíč, hodnota.

Lokalizace JavaScriptu

Jak lokalizovat jazyk, který nedovoluje importovat soubory?

Ideální varianta je držet lokalizované řetězce mimo JavaScriptový kód. Jedna možnost je využít pro JavaScript JSP soubory a použít standardní postupy pro lokalizaci JSP stránek. To ovšem není vždy možné a ani výkon nebude optimální.

IBM využívá hezký postup, kdy je pro každý jazyk vygenerována jedna varianta JavaScriptového souboru. Musíme upravit zdrojové soubory, takže z

```
alert("Hello world");
```

vznikne

```
alert ("@hello.message@");
```

A pak využijeme ant a properties soubory.

```
<loadfile srcfile="${src.file}" property="${src.file.replaced}">
<filterchain>
<filterreader classname="org.apache.tools.ant.filters.ReplaceTokens">
<param type="propertiesfile" value="sample.properties"/>
</filterreader>
</filterchain>
<filterchain>
</loadfile>
```

Další varianta která se nabízí je použít AJAX pro lokalizaci řetězců. Ale to je příliš náročné na přenos a zbytečně komplikované.

Ukládání datových atributů v jazykových variantách

Předpokládejme, že naše aplikace potřebuje ukládat název článku v několika jazycích. V případě, že neplánujeme více než několik málo jazyků, můžeme pro každý jazyk vytvořit jeden datový atribut u objektu, do kterého budeme ukládat informace. Například `heading_cs` a `heading_en`. Pokud plánujeme lokalizovat do většího množství jazyků, vzniká potřeba dynamicky generovat strukturu pro data. Můžeme dynamicky vytvářet tabulky pro atribut `Heading`, každá bude mít příponu `_locale`, kde `locale` bude nahrazeno zkratkou daného jazyka. Poté, kdykoli budeme vypisovat název článku, podíváme se, jestli existuje tabulka s názvem pro daný jazyk a vypíšeme data z ní, jinak použijeme výchozí.

Lokalizace šablon

Šablony jsou značným ulehčením lokalizace. Existuje jediná varianta zdrojového kódu, ve kterém jsou pak určité značky nahrazovány za skutečná data.

JSP

Pro lokalizaci JSP stránek se využívají taglibs. Taglib je knihovna tagů(značek), které jsou při překladu JSP stránky do Java servletu interpretovány a vyhodnoceny. V tomto okamžiku je lokalizační značka vyhodnocena, použije se Bundle specifikovaný v hlavičce stránky a daný výraz nahrazen překladem z Bundlu.

Velocity

K lokalizaci Velocity šablon je používáno více způsobů. Například označení řetězců k lokalizaci speciálními značkami.

```
$-Text k lokalizaci-$
```

Poté parserem projdeme všechny soubory se šablonami, a každý takto označený výskyt klíče označíme unikátním indexem

```
$-Text k lokalizaci/xtz-$
```

Značka xtz je jednoznačný identifikátor v celém systému. Na základě tohoto identifikátoru poté probíhá při generování jednotlivých jazykových verzí šablon náhrada za lokalizovaný řetězec.

Další možná varianta je využít metod z Javy, protože při parsování šablon můžeme používat objekty i jejich metody. Pokud tedy vložíme do šablony objekt Bundle, a poté budeme volat například:

```
$bundle.getString(„klíč“)
```

Toto volání lze ještě zjednodušit využitím maker, více viz sekce implementace.

Lokalizace grafiky

Grafiku můžeme lokalizovat obdobně jako jiná data. Název souboru můžeme uložit jako lokalizovatelný řetězec a pro každý jazyk pozměnit jeho název přímo při překladu. Nebo je tento proces možné dělat automaticky, pro každý soubor s obrázkem připojovat za jeho jméno příponu, nebo ho hledat v jiném adresáři.

6.2 O implementovaném softwaru

V rámci mé bakalářské práce jsem se rozhodl implementovat jednoduchou webovou aplikaci pro správu a sdílení odkazů a souborů. Představa je taková, že software bude umět velmi dynamicky reagovat na změny a přidávání jazyků, nebude problém za chodu pomocí webového rozhraní administrovat jak překlady, tak samotné jazyky.

Platforma

Pro vývoj aplikace jsem zvolil jako platformu Ubuntu. Ubuntu je linuxová distribuce pro pracovní stanice, založená na Debian GNU/Linux. Pro tuto platformu jsou dostupné veškeré použité technologie, tedy Java, PostgreSQL i Tomcat. Tyto technologie je možné používat i na platformě Microsoft Windows, neměl by tedy být problém projekt přesunout tam.

Programovací jazyk

Jako programovací jazyk byl od začátku použit jazyk Java. Jednak proto, že po celou dobu studia na univerzitě je tento jazyk upřednostňován, jednak kvůli jeho rozšířenosti a multiplatformnosti. Na jednoduché aplikace tohoto typu se běžně používá jazyk PHP. Umožňuje rychlý vývoj a snadné nasazení. Bohužel má mnoho nevýhod, které vyplývají z jeho jednoduchosti. Rád bych nastínil několik důvodů, proč si myslím, že i na malé aplikace není Java „kanón na vrabce“.

Striktní jazyk: Java je velmi přísná, co se týče datových typů. Nedovolí implicitně přetypovat čísla v případě, že bychom ztratili přesnost. Většinu chyb odhalí již překladač

Výjimky: propracovaný systém výjimek umožňuje snadno zpracovat chyby vzniklé při běhu aplikace

JavaDoc: dokumentace pomocí komentářů přímo ve zdrojovém kódu, následná možnost vygenerování HTML souborů s touto dokumentací

Aplikační kontejner

Software je postaven jako webová aplikace, běžící v aplikačním kontejneru Tomcat 6. Tomcat je standardem pro webové aplikace, je velmi dobře začleněn do vývojového

prostředí Netbeans, ve kterém probíhal vývoj a ladění aplikace. Díky standardizaci by však neměl být problém vzít celou aplikaci a pustit ji pod jiným servlet kontejnerem.

Databáze

Aplikace uchovává data v PostgreSQL databázi. PostgreSQL byla zvolena díky mým zkušenostem s ní, BSD licenci, která je velmi volná a dobré rozšiřitelnosti databáze.

6.3 Detaily implementace

Objektově relační mapování

Protože Java je čistě objektový jazyk a PostgreSQL relační databáze, rozhodl jsem se použít objektově-relační mapování, abych co nejvíce zjednodušil práci s databází. Vzhledem k předchozím zkušenostem a snadné implementaci jsem volil Hibernate framework. Práce s tímto frameworkem je velmi jednoduchá. Vytvoří se třída splňující požadavky na JavaBean, tedy bezparametrický konstruktor, gettery a settery k proměnným třídy. K této třídě se vytvoří mapovací soubor XML, kde říkáme, v jaké tabulce budou uloženy proměnné, jak se budou jmenovat sloupce, ale i jaké bude datové mapování. Jednoduchý příklad by vypadal třeba následovně:

Java třída Language, nesoucí informaci o jazyku a všech slovech přiřazených tomuto jazyku.

```
public class Language {
    private int id;
    private String locale;
    private Set<Word> words;
    public Language() {
    }
    public int getId() {
        return id;
    }
    private void setId(int id) {
        this.id = id;
    }
    public String getLocale() {
        return locale;
    }
    public void setLocale(String locale) {
        this.locale = locale;
    }
    public Set<Word> getWords() {
        return words;
    }
    public void setWords(Set<Word> words) {
```

```

        this.words = words;
    }
}

```

Soubor Language.hbm.xml říká, že tabulka se bude jmenovat languages, jaké budou její sloupce, a pak datové mapování, kdy třída Language obsahuje set složený z objektů typu Word. Vazby mezi Language a Word zajišťuje tabulka language_words.

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class lazy="false" name="cz.dvorak.zalozky.beans.lang.Language"
table="languages">
    <id column="id_language" name="id">
      <generator class="native"/>
    </id>
    <property name="locale"/>
    <set lazy="false" name="words" table="language_words">
      <key column="id_language"/>
      <many-to-many class="cz.dvorak.zalozky.beans.lang.Word"
column="id_word"/>
    </set>
  </class>
</hibernate-mapping>

```

Šablonovací systém

Nejjednodušší způsob výstupu webové aplikace je čistý výpis do OutputWriteru. To je ale naprosto nepoužitelné, neupravitelné. O něco lepší přístup je použít JSP stránky. Jenže tvorba JSP stránek je do značné míry komplikovaná a nesplňuje požadavky na jednoduchost zobrazovací vrstvy a možnost editace někým, kdo není zběhlý v programování v Javě. Proto jsem použil šablonovací systém Velocity. Ten umožňuje velmi volný přístup k Java objektům. Přenos dat mezi aplikací a šablonou probíhá tak, že aplikace vygeneruje data, která vloží do kontextu. Kontext je ve skutečnosti jen mapa, kde je pod zvoleným názvem dostupný objekt. V šablonách není třeba řešit datový typ objektu ani nic podobného. Velocity umožňuje přistupovat k objektům velmi jednoduše pomocí getterů. Šablonovací systém mi následně velmi zjednoduší práci při lokalizaci softwaru.

Postup lokalizace

V Javě se všechno, co se týče lokalizace, točí okolo jediného pojmu ResourceBundle. Připomeňme si tedy, jak ResourceBundle funguje. Pokaždé, když potřebuji získat lokalizovaný řetězec, požádám o něj Bundle a nemusím se dále starat, jak a kde překlád

získá. Lokalizované řetězce jsou ukládány v souborech s koncovkou .properties a názvem souboru skládajícím se z názvu Bundlu a identifikátoru locale. Soubor s překlady navíc musí být kódován pomocí ASCII. Ve zdrojovém kódu Javy pak práce vypadá asi takto :

```
ResourceBundle myResources = ResourceBundle.getBundle("MyResources",  
currentLocale);
```

Při takovémto zápisu pak Java bude postupně hledat tyto soubory :

```
messages_cs_CZ.properties  
messages_cs.properties  
messages_en_US.properties  
messages_en.properties  
messages.properties
```

Pokud nějaký najde, bude v něm následně hledat klíče které budeme po Bundlu požadovat. Takový postup jistě není špatný pro desktopové aplikace. Ale webové prostředí má jiné požadavky a možnosti. Už jen sama editace souborů s překlady je na webu problematická. Proto budeme implementovat vlastní ResourceBundle, který umožní překlady nahrát z databáze. I chování ke klíčům, které neexistují, bude odlišné. Abych při lokalizaci nemusel řešit, kde jsem které klíče zapomněl zaevidovat, postup bude takový, že pokud aplikace bude žádat o klíč, který dosud neexistuje, zaeviduje se nový a bude čekat na překlad. Nevýhoda tohoto postupu je, že nemám ihned dostupnou celou množinu všech klíčů k překladu. Tuto nevýhodu ale bohatě převáží pohodlnost práce, kde se nestarám o klíče, prostě píšu šablony a při jejich prvním zobrazení se klíče samy vygenerují.

Začněme tedy implementací vlastního ResourceBundle. Měl by umět získávat slova z databáze a mít možnost přidat nové, pokud nenalezne v databázi odpovídající klíč. Samotnou práci nad databází vyčleníme do jiné třídy, DatabaseDictionaryBuilder. Ta umí jen vygenerovat slovník z databáze a poradit si v případě, že jí DatabaseResourceBundle sdělí, že slovo nenašel, případně vyrovnat se se stavem, kdy Bundle požaduje jazyk, který není vůbec evidován. Podstatná pro chod celého systému práce s Bundle je třída Controls. Ta umožňuje samotné natažení vlastního ResourceBundlu. Podrobnosti viz zdrojové kódy a JavaDoc dokumentace.

Databázový model, řešení dynamického přidávání jazyků

Díky vhodně nakonfigurovanému objektově-relačnímu mapování se nemusíme starat

o práci s databází, vše probíhá přes Hibernate. To nám na základě locale poskytne odpovídající objekt Language, který s sebou nese i veškerá slova. Dynamika lokalizace je dána tím, že aplikace umožňuje kdykoli za běhu vytvářet nové objekty Language a přiřazovat jim nová slova. Ta jsou na pozadí mapována a ukládána do relační databáze. Pokud chceme získat nový jazyk, stačí aplikaci požádat o překlad nějakého slova v tomto jazyce. Automaticky bude vytvořen jazyk i slovo.

Lokalizace pomocí Velocity šablon

Jak již bylo uvedeno dříve, Velocity umožňuje přistupovat k metodám objektů a vypsat jejich návratovou hodnotu. Tohoto postupu využijeme při lokalizaci. Šablony mohou získávat objekty z kontextu. Při každém zpracování šablony tedy do kontextu vložíme Bundle s příslušným jazykem. Tento Bundle pak budeme volat pro každý lokalizovaný řetězec. Aby byla práce ještě příjemnější, využijeme další možnosti, kterou Velocity poskytuje, tedy Velocity makro.

Definujeme tedy následující makro:

```
#macro(M $key)
$bundle.getString($key)
#end
```

Kód říká, že definujeme makro se jménem M, které má jeden vstupní parametr s názvem key. Tento parametr použijeme jako parametr pro metodu getString(String key). Tato metoda požádá Bundle o nalezení překladu pro locale, které obsahuje.

V šabloně pak bude použití následovné:

```
<h1>#M(„Nadpis“)</h1>
<ul>
  <li>#M(„První položka“)</li>
  <li>#M(„Druhá položka“)</li>
</ul>
```

Zápis pomocí makra mi přijde přece jen příjemnější, než například zápis v JSP :

```
<h1><i18n:message key="nadpis" /></h1>
<ul>
  <li><i18n:message key="První položka" /></li>
  <li><i18n:message key="Druhá položka" /></li>
</ul>
```

Výhodou je, že pokud slovo v Bundle neexistuje, vypíše se přesně to, co bylo zadáno.

Přepínání jazyků

Jazyk, kterým budou stránky vypsány, se získává v session. Stejným způsobem můžeme jazyk nastavit my.

```
private Locale discoverLocale(HttpServletRequest request) {  
    if (request.getParameter(LOCALE_KEY) != null) {  
        String localeString = request.getParameter(LOCALE_KEY);  
        Locale locale = new Locale(localeString);  
        request.getSession(true).setAttribute("userLocale",  
locale);  
        return locale;  
    } else if (request.getSession().getAttribute("userLocale") !=  
null) {  
        return (Locale)  
request.getSession().getAttribute("userLocale");  
    } else {  
        return request.getLocale();  
    }  
}
```

Pokud tuto metodu budeme volat při každém zpracování požadavku na serveru, zajistíme, že pokaždé, když v parametru přijde proměnná locale, změníme jazyk.

Administrace překladů

Administrace překladů probíhá kompletně přes webové rozhraní. Uživatel má možnost zvolit ze všech dostupných jazyků v aplikaci. Po volbě jednoho konkrétního jazyka se zobrazí tabulka všech slov vyžadovaných tímto jazykem. Pokud již mají nějaký překlad, vypíše se i ten. Každý překlad je možné editovat. Při stisku tlačítka „uložit“ se vytvoří z dat odeslaných na server dvojice klíč-překlad, každá z nich je postupně kontrolována, pokud obsahuje novou nebo změněnou hodnotu, je uložena do databáze. Na konci tohoto procesu se vymaže cache ResourceBundlu, tím je zajištěno nové nahrání veškerých potřebných dat z databáze a tím i aktuálnost všech překladů.

Uživatelské rozhraní je samozřejmě opět lokalizováno do všech dostupných jazyků(ilustace 9.).



Bookmarks

Choose language : en

Logged in

Tomáš Dvořák

- [Options](#)
- [Help](#)
- [Localize](#)
- [Logout](#)

Available languages : [cs](#) [de](#) [en](#) [fr](#) [ru](#)

You are editing language : en

Sdílené projekty
Pouze čtení
přidat složku
Heslo
Nashledanou...
Vložit složku
Vložit zalozku
Čtení, zápis
Popis projektu
Vložit soubor
Přihlásit
Nadpis
přidat soubor
E-mail

Shared projects
Add folder
Pass
Insert folder
Insert bookmark
Insert file
Login
Heading
Add file

Ilustrace 10: Editační rozhraní pro lokalizaci, přeložené do angličtiny

7. Závěr

Cílem této práce bylo nabídnout současné trendy v lokalizaci IT produktů. Zaměřil jsem se na produkty společností Google a Microsoft, jako největších firem v oblasti internetu a stolních počítačů. Obě společnosti nabízejí nástroje pro strojový překlad textu v různých situacích. Popsal jsem tedy tyto nástroje a jejich použití. Dále jsem se věnoval lokalizaci obecně, lokalizačnímu procesu z pohledu samotného návrhu a časovému rozložení projektu lokalizace. Tato část práce by měla posloužit jako seznam kroků a jejich vhodný sled při vývoji softwaru, který bude lokalizován.

Podrobněji je rozebráno použití Gettextu. Gettext je nástroj pro lokalizaci zdrojových kódů, především PHP a jazyka C. Práce s ním je velmi snadná a přehledná. Druhý nástroj popsáný v teoretické části je OmegaT. OmegaT je software s otevřeným zdrojovým kódem, podobně jako Gettext. Zaměřuje se však na překlad jiného spektra zdrojů. Jde především o HTML stránky, texty dokumentace, obsah psaný pomocí textových editorů, jako je MS Word nebo OpenOffice Writer. Umožňuje však samozřejmě překládat i texty použité právě Gettextem.

V druhé části práce jsem zvolil k lokalizaci svůj vlastní projekt Záložky. Záložky jsou webová aplikace umožňující uživatelům přidávat odkazy na internetové stránky, sdílet a komentovat tyto odkazy s ostatními registrovanými uživateli. Lokalizaci tohoto projektu jsem se rozhodl implementovat pomocí jazyka Java a jím nabízených lokalizačních postupů. Značná část těchto postupů však nesplňovala moje požadavky na správu lokalizovaných textů a možnosti jejich ukládání, proto jsem patřičné postupy programoval sám. Jednalo se především o ukládání překladů v databázi, jejich automatické načtení do uživatelského rozhraní v okamžiku, kdy existuje novější verze překladů a celková dynamika práce s lokalizovanými texty. Vytvořil jsem webové rozhraní ke slovníku a editaci překladů.

Lokalizace splnila má očekávání na rychlost a dynamiku. Je však nutné počítat s tím, že v mé aplikaci je lokalizováno pouze několik desítek řetězců. Při opravdovém nasazení na projektu většího rozsahu by bylo zřejmě nutné upustit od některých konkrétních aspektů, jako je automatická aktualizace překladů celé aplikace po změně překladu. Případně přistoupit k vygenerování jazykových variant pro každou stránku.

Zdroje

Internetové zdroje

www.wikipedia.org – všeobecná encyklopedie, použita zejména pro informace o licencích programů, výpisu lokalizačních nástrojů

www.root.cz – server o programování a linuxu, detaily o funkci gettext

www.google.com/translate – rozcestník společnosti Google v oblasti lokalizace

www.livetranslate.com – rozcestník společnosti Microsoft v oblasti lokalizace

java.sun.com – Portál společnosti Sun pro jazyk Java, informace o lokalizaci a detailech implementace Locale a ResourceBundle

www.donpedrowebdesign.netfirms.com/choose_color.html – teorie barev a jejich význam v jednotlivých kulturách

www.rotowatch.com/ – Lokalizace JavaScriptu, návod jak pomocí Antu vyřešit lokalizace JavaScriptu

Tištěné zdroje

Bert Esselink – A Practical Guide to Localization

Ucelený přehled o lokalizaci, detaily postupů a popisy jednotlivých aspektů lokalizace

Ke stažení například na adrese http://www.ebook3000.com/A-Practical-Guide-to-Localization_8496.html

Bytelevel – Website Globalization Report 2007

Průzkum a statistiky z oblasti globalizace webu

ke stažení například na adrese

<http://www.marketingsherpa.com/Reports/WebsiteGlobalization.pdf>

Přílohy

Příloha 1: Tabulka locales, ukázka výpisu v daném locale

Příloha 2: Snímky obrazovek programu

Příloha 3: Přehled lokalizačních nástrojů

Příloha 4: Snímek obrazovky běhu aplikace OmegaT

Obsah CD

Složka **zdroje**: obsahuje kompletní projekt aplikace NetBeans. Import tohoto projektu do NetBeans je asi nejsnazší varianta jak aplikaci spustit.

Složka **tisk**: obsahuje tuto bakalářskou práci ve formátu PDF určenou pro tisk a prohlížení v počítači.

Složka **dokumentace**: obsahuje soubor readme.txt popisující jednotlivé kroky nutné ke spuštění aplikace, potřebné knihovny a nainstalované aplikace. Dale obsahuje krátký uživatelský manuál a dokumentaci JavaDoc vygenerovanou ze zdrojových kódů.

Složka **spustitelne**: obsahuje soubor zalozky.war. Je to soubor spustitelný v aplikačním kontejneru Tomcat.

Příloha 1: Tabulka locales

Locale	Jazyk	Země	Datum (Dlouhý formát)	Datum(krátké)	Měna
ja_JP	Japanese	Japan	2009/04/22	09/04/22	¥100
es_PE	Spanish	Peru	22 de abril de 2009	22/04/09	S/99,90
en	English		April 22, 2009	4/22/09	¤99,90
ja_JP_JP	Japanese	Japan	H21.04.22	H21.04.22	¥100
es_PA	Spanish	Panama	22 de abril de 2009	04/22/09	B99,90
sr_BA	Serbian	Bosnia and Herzegovina	22. април 2009.	09-04-22	KM. 99,90
mk	Macedonian		22. април 2009	22.4.09	¤ 99,90
es_GT	Spanish	Guatemala	22 de abril de 2009	22/04/09	Q99,90
ar_AE	Arabic	United Arab Emirates	22 2009 أبريل	22/04/09	99.9 د.إ.
no_NO	Norwegian	Norway	22. april 2009	22.04.09	kr 99,90
sq_AL	Albanian	Albania	2009-04-22	09-04-22	Lek99,9
bg	Bulgarian		Сряда, 2009, Април 22	09-4-22	¤ 99,90
ar_IQ	Arabic	Iraq	22 2009 أبريل	22/04/09	99.9 د.ع.
ar_YE	Arabic	Yemen	22 2009 أبريل	22/04/09	99.9 ر.ي.
hu	Hungarian		2009. április 22.	2009.04.22.	¤ 99,90
pt_PT	Portuguese	Portugal	22 de Abril de 2009	22-04-2009	99,90 €
el_CY	Greek	Cyprus	22 Απρίλιος 2009	22/04/2009	€99,90
ar_QA	Arabic	Qatar	22 2009 أبريل	22/04/09	99.9 ر.ق.
mk_MK	Macedonian	Macedonia	22. април 2009	22.4.09	Den 99,9
sv	Swedish		den 22 april 2009	2009-04-22	¤ 99,90
de_CH	German	Switzerland	22. April 2009	22.04.09	SFr. 99,90
en_US	English	United States	April 22, 2009	4/22/09	\$99,90
fi_FI	Finnish	Finland	22. huhtikuuta 2009	22.4.2009	99,90 €
is	Icelandic		22. apríl 2009	22.4.2009	¤ 99,90
cs	čeština		22. duben 2009	22.4.09	¤ 99,90
en_MT	English	Malta	22 April 2009	22/04/2009	€99,90
sl_SI	Slovenian	Slovenia	Sreda, 22 april 2009	22.4.09	€ 99,9
sk_SK	Slovak	Slovakia	Streda, 2009, april 22	22.4.2009	99,90 Sk
it	Italian		22 aprile 2009	22/04/09	¤ 99,90
tr_TR	Turkish	Turkey	22 Nisan 2009 Çarşamba	22.04.2009	99,90 Y TL
zh	Chinese		2009年4月22日	09-4-22	¤ 99,90
th	Thai		เมษายน 22, 2009	4/22/09	¤ 99,90
ar_SA	Arabic	Saudi Arabia	22 2009 أبريل	22/04/09	99.9 ر.س.
no	Norwegian		22. april 2009	22.04.09	¤ 99,90
en_GB	English	United Kingdom	22 April 2009	22/04/09	£99,90
sr_CS	Serbian	Serbia and Montenegro	22.04.2009.	22.4.09.	CSD 99,90
lt	Lithuanian		Trečiadienis, 2009, Balandžio 22	09.4.22	¤ 99,90
ro	Romanian		22 aprilie 2009	22.04.2009	¤ 99,90
en_NZ	English	New Zealand	22 April 2009	22/04/09	\$99,90
no_NO_NY	Norwegian	Norway	22. april 2009	22.04.09	kr 99,90
lt_LT	Lithuanian	Lithuania	Trečiadienis, 2009, Balandžio 22	09.4.22	99,9 Lt
es_NI	Spanish	Nicaragua	22 de abril de 2009	04-22-09	\$C99,90
nl	Dutch		22 april 2009	22-4-09	¤ 99,90
ga_IE	Irish	Ireland	22 Aibreán 2009	22/04/2009	€99,90
fr_BE	French	Belgium	22 avril 2009	22/04/09	99,90 €
es_ES	Spanish	Spain	22 de abril de 2009	22/04/09	99,90 €

es_DO	Spanish	Dominican Republic	22 de abril de 2009	04/22/09	RD\$99.90
fr_CH	French	Switzerland	22. avril 2009	22.04.09	SFr. 99.90
hi_IN	Hindi	India	२२ अप्रैल, २००९	२२/४/०९	₹ ९९.९०
es_VE	Spanish	Venezuela	22 de abril de 2009	22/04/09	BsF.99.90
ar_BH	Arabic	Bahrain	22 أبريل, 2009	22/04/09	د.ب. 99.9
en_PH	English	Philippines	April 22, 2009	4/22/09	Php99.90
ar_TN	Arabic	Tunisia	22 أبريل, 2009	22/04/09	د.ت. 99.9
fi	Finnish		22. huhtikuuta 2009	22.4.2009	¤ 99,90
de_AT	German	Austria	22. April 2009	22.04.09	€ 99,90
es	Spanish		22 de abril de 2009	22/04/09	¤99,90
nl_NL	Dutch	Netherlands	22 april 2009	22-4-09	€ 99,90
es_EC	Spanish	Ecuador	22 de abril de 2009	22/04/09	\$99,90
zh_TW	Chinese	Taiwan	2009年4月22日	2009/4/22	NT\$99.90
ar_JO	Arabic	Jordan	22 نيسان, 2009	22/04/09	د.أ. 99.9
be	Belarusian		серада, 22, красавіка 2009	22.4.09	¤ 99,90
is_IS	Icelandic	Iceland	22. apríl 2009	22.4.2009	100, kr.
es_CO	Spanish	Colombia	22 de abril de 2009	22/04/09	\$99,90
es_CR	Spanish	Costa Rica	22 de abril de 2009	22/04/09	C99.90
es_CL	Spanish	Chile	22 de abril de 2009	22-04-09	Ch\$100
ar_EG	Arabic	Egypt	22 أبريل, 2009	22/04/09	ج.م. 99.9
en_ZA	English	South Africa	22 April 2009	2009/04/22	R 99.90
th_TH	Thai	Thailand	22 เมษายน 2552	22/4/2552	฿99.90
el_GR	Greek	Greece	22 Απρίλιος 2009	22/4/2009	99,90 €
it_IT	Italian	Italy	22 aprile 2009	22/04/09	€ 99,90
ca	Catalan		22 / abril / 2009	22/04/09	¤ 99,90
hu_HU	Hungarian	Hungary	2009. április 22.	2009.04.22.	99,9 Ft
fr	French		22 avril 2009	22/04/09	99,90 ¢
en_IE	English	Ireland	22 April 2009	22/04/09	€99.90
uk_UA	Ukrainian	Ukraine	22 квітня 2009	22.04.09	99,9 грн.
pl_PL	Polish	Poland	22 kwiecień 2009	22.04.09	99,9 zł
fr_LU	French	Luxembourg	22 avril 2009	22/04/09	99,90 €
nl_BE	Dutch	Belgium	22 april 2009	22/04/09	99,90 €
en_IN	English	India	22 April, 2009	22/4/09	Rs.99.90
ca_ES	Catalan	Spain	22 / abril / 2009	22/04/09	€ 99,90
ar_MA	Arabic	Morocco	22 أبريل, 2009	22/04/09	د.م. 99.9
es_BO	Spanish	Bolivia	22 de abril de 2009	22-04-09	B\$99.90
en_AU	English	Australia	22 April 2009	22/04/09	\$99.90
sr	Serbian		22.04.2009.	22.4.09.	¤ 99,90
zh_SG	Chinese	Singapore	22 四月 2009	22/04/09	S\$99.90
pt	Portuguese		22 de Abril de 2009	22-04-2009	¤ 99,90
uk	Ukrainian		22 квітня 2009	22.04.09	¤ 99,90
es_SV	Spanish	El Salvador	22 de abril de 2009	04-22-09	C99.90
ru_RU	Russian	Russia	22 Апрель 2009 г.	22.04.09	99,9 руб.
ko_KR	Korean	South Korea	2009년 4월 22일 (수)	09. 4. 22	₩100
vi	Vietnamese		Ngày 22 tháng 4 năm 2009	22/04/2009	¤ 99,90
ar_DZ	Arabic	Algeria	22 أبريل, 2009	22/04/09	د.ج. 99.9
vi_VN	Vietnamese	Vietnam	Ngày 22 tháng 4 năm 2009	22/04/2009	99,90 đ

en_CA	English	Canada	April 22, 2009	22/04/09	\$99.90
de_DE	German	Germany	22. April 2009	22.04.09	99,90 €
ga	Irish		2009 Aibreán 22	09/04/22	¤ 99.90
de_LU	German	Luxembourg	22. April 2009	22.04.09	99,90 €
de	German		22. April 2009	22.04.09	¤ 99,90
es_AR	Spanish	Argentina	22 de abril de 2009	22/04/09	\$99,90
sk	Slovak		Streda, 2009, apríl 22	22.4.2009	¤ 99,90
ms_MY	Malay	Malaysia	22 April 2009	22/04/2009	RM99.90
hr_HR	Croatian	Croatia	2009. travanj 22	2009.04.22	Kn 99,9
en_SG	English	Singapore	April 22, 2009	4/22/09	\$99.90
da	Danish		22. april 2009	22-04-09	¤ 99,90
mt	Maltese		22 ta' April 2009	22/04/2009	¤ 99.90
pl	Polish		22 kwiecień 2009	09-04-22	¤ 99,90
ar_OM	Arabic	Oman	22 أبريل 2009	22/04/09	99.9 ر.ع.
tr	Turkish		22 Nisan 2009 Çarşamba	22.04.2009	99,90 ¤

Příloha 2: Snímky obrazovek aplikace Záložky



Bookmarks

Choose language : en

Logged in
Tomáš Dvořák

- [Options](#)
- [Help](#)
- [Localize](#)
- [Logout](#)

Private projects :

- [Třetí](#)
- [Standartní](#)
- [Druhý - 2](#)

Shared projects :

- [Standartní](#)

[Add project](#) - [Add folder](#) - [Add file](#) - [Comment](#)

Škola :

- <http://www.kiv.zcu.cz> -- KIV >>
- http://www.kiv.zcu.cz/studies/temata/popis.php?id_popis=1323 -- Bakalářka >>
- <http://portal.zcu.cz> -- Portál zcu >>
- <http://students.cz> -- Students.cz >>

Test :

- <http://www.google.com> -- Google >>
- <http://localhost:8084/zalozky/screens/Profile.vm#> -- Test >>

Příloha 2, ilustrace 1 : Aplikace záložky přeložená do angličtiny



Záložky

Zvolte jazyk : cs

Přihlášen
Tomáš Dvořák

- [Nastavení](#)
- [Nápověda](#)
- [Lokalizovat](#)
- [Odhlásit](#)

Dostupné jazyky : cs de en fr ru

Právě editujete jazyk : en

Odhlásit

Nastavení práv

Sdílené projekty

Odkaz

Absolutní práva (číst, zapisovat, mazat)

Nashledanou...

Práva k projektu

Zvolte jazyk

Vložit soubor

Nastavení

Záložky

text

Příloha 2, ilustrace 2 : Správa lokalizovaných řetězců



Záložky

Zvolte jazyk : CS

Přihlášení

Přezdívka

Heslo

Registrace

Nemáte ještě vlastní účet? [Zaregistrujte se](#)

Proč používat záložky?

Stalo se Vám někdy že jste si potřebovali poznamenat nějaký odkaz z počítače ve škole, kavárně, abyste se na něj mohli doma podívat? Psát ho na papír snad nikoho nenapadne, ukládat do textového souboru je pracné, posílat si ho mejlem není pohodlné. Tady si pouze poznamenáte adresu a k ní popis, a vše zůstane dostupné odkudkoli

Registrace

Povinné údaje jsou pouze jméno a heslo. E-mail vyplňte jen v případě, že si chcete nechat zaslat nové heslo v případě ztráty hesla aktuálního.

Přihlášení

Je nutné pro ověření vaší identity. Pokud se Vám nedaří přihlásit, je pravděpodobné, že jste zadali jméno nebo heslo jinak než při registraci. Velká a malá písmena jsou důsledně rozlišována! A to i ve jméně.

(c)2009 Tomáš Dvořák

Příloha 2, ilustrace 3 : Úvodní strana aplikace



Záložky

Zvolte jazyk : CS

Přihlášen

Tomáš Dvořák

- [Nastavení](#)
- [Nápověda](#)
- [Lokalizovat](#)
- [Odhlásit](#)

Editace záložky

Popis

Odkaz

(c)2009 Tomáš Dvořák



Příloha 2, ilustrace 4 : Editace jedné záložky

Příloha 3: Lokalizační nástroje

Nástroj	Podporované formáty	OS	Cena	Licence
ForeignDesk	HTML, zdrojové kódy v C, Java Source Code, Microsoft Help File Sources (HPJ, HHC, HHK, CNT), Trados,	Windows		Open Source
Okapi Framework	PO, Windows RC, TMX, Wordfast, Trados, Java Properties, Regular-expression-based text, Illustrator, INX, ResX, Table-type files, XML	Windows (.NET)		LGPL
OmegaT	HTML, DocBook, Plain Text, PO, JavaHelp, Java Resource Bundles, OpenOffice.org, StarOffice	multiplatformní (Java)		Open Source
Transolution	HTML, StarOffice/OpenOffice.org, XLIFF, DOCBOOK	multiplatformní (Python)		GPL
AidTransStudio	OpenOffice.org, MS Word Xml	Windows (.NET)	Základní: zdarma	vlastní řešení
Cafetran	HTML, XML, OpenOffice.org, AbiWord, Kword, MS Word	multiplatformní (Java)	180 euro	
Déjà Vu (DVX)	XML, Plain Text, OpenOffice.org, Adobe FrameMaker, Adobe PageMaker, ASP, Interleaf/Quicksilver, InDesign, Help Content, SGML, MS Access, MS Excel, MS PowerPoint, MS Word, QuarkXPress, RTF, Resource files, C/C++/Java source files, Java Properties, JavaScript, VBScript, GNU gettext	Windows	490 euro	vlastní řešení
Heartsome Translation Suite	HTML/XHTML, XML, Plain Text, OpenOffice.org, StarOffice, AbiWord, PO/POT (GNU Gettext), SVG, Adobe FrameMaker (MIF), Adobe InDesign, DocBook, DITA, Java Properties, JavaScript, RTF, Tagged RTF, Trados TTX, MS Office 2003 XML, ResX (Windows .NET Resources), RC (Windows C/C++ Resources), MS Office 2007 (beta)	multiplatformní (Java)	60-360 euro	vlastní řešení
MemoQ	HTML, prostý text, MS Word (plus RTF a další dokumenty Wordu), Excel, PowerPoint, Trados TTX, vlastní dvojjazyčný formát	Windows	4Free: Freeware Translator Pro: 390 euro LSP 5: 1490 euro	vlastní řešení
MetaTaxis	HTML, XML, Resource files MS Word (všechny textové soubory, které lze importovat do MS Wordu), MS Excel, MS PowerPoint, Adobe FrameMaker, Adobe PageMaker, QuarkXPress	Windows	Lite: zdarma Pro: 98 eur .NET/Office: 138 eur	vlastní řešení
Rainbow	HTML, XHTML, Scripts, Photoshop, etc.	Windows (.NET)	Freeware	vlastní řešení
SDLX	RTF, MS Word, HTML, XML, SGML, další webové soubory, MS PowerPoint, MS Excel, FrameMaker, CopyFlow, Trados RTF, STF, TTX, soubory Windows RC, soubory Java Properties, InDesign, Quark	?	?	
STAR Transit	Text ANSI / ASCII / Unicode pro Windows, Text pro Apple Macintosh, Corel WordPerfect, HTML, XML (ASP.NET, ASP, JSP, XSL), SGML, SVG (Scalable Vector Graphics), MS Word, MS Excel, MS PowerPoint, RTF y RTF for WinHelp, RC, QuarkXPress, Adobe FrameMaker, Adobe PageMaker, Interleaf /Quicksilver, Adobe InDesign, XGate para QuarkXPress, AutoCAD	Windows		vlastní řešení
Trados Translators Workbench	Po instalaci automaticky přidává šablonu do MS Wordu. Pro soubory Excelu, PowerPointu by se měl používat Tag Editor instalovaný spolu s nástrojem. Podporuje i další formáty, jakými jsou např. OpenOffice, INX, HTM(L), ASP(X), XML, RTF, XLIFF, DLL, Java Properties, QuarkXPress, MIF a další.	Windows 2000, XP, Vista 32, MacOS	995 \$	
Wordfast	MS Word	Microsoft Office Word macro	180 euro	vlastní řešení
WordFisher	MS Word	WordBasic/MS Office Word macro	Free Licence	
Similis	HTML, PDF, Word, Trados	Windows	295 euro (monoposte)	vlastní řešení
Open Language Tools	HTML/XHTML, XML, DocBook SGML, ASCII, StarOffice/OpenOffice/ODF, .po (gettext), .properties, java (ResourceBundle), .msg/.tmsg (catgets)	multiplatformní (Java)	zdarma	CDDL
poEdit software		multiplatformní	zdarma	
Anaphraseus	Speciálně pro OpenOffice. Po instalaci automaticky přidává šablonu do textového editoru.	Multiplatformní, vyžaduje OpenOffice 2.0 a novější	zdarma	

Příloha 4: OmegaT

