
Specifikace požadavků

pro

Systém pro editaci podmínek ConEdit

Verze 1.2

Autor: Vladimír Rovenský

MFF UK

17.11.2009

Obsah

1. Úvod	1
1.1 Účel dokumentu.....	1
1.2 Úvod do problematiky	1
1.3 Popis modelu	1
1.4 Stručný popis formátu souboru podmínek	2
1.5 Příklad podmínky v souboru.....	3
1.6 Reference	3
2. Celkový popis	3
2.1 Popis aplikace.....	3
2.2 Běhové prostředí.....	3
2.3 Dokumentace	4
2.4 Negativní vymezení	4
3. Funkce aplikace.....	4
3.1 Načtení souboru podmínek.....	4
3.2 Přidávání, odebírání a editace podmínek	5
3.3 Online aktualizace souboru podmínek	5
3.4 Odstraňování redundantních podmínek.....	5
4. GUI.....	6
5. Nefunkční požadavky	10
5.1 Lokalizace.....	10
5.2 Minimální HW a SW požadavky.....	10
5.3 Výkon	10
5.4 Další požadavky	10

Historie verzí

Jméno	Datum	Popis změny	Verze

1. Úvod

1.1 Účel dokumentu

Tento dokument slouží coby specifikace požadavků na aplikaci ConEdit spravující vstupní soubor podmínek projektu *Sense*, jenž na základě tohoto souboru rozhoduje o smysluplnosti českých vět. Dokument stanovuje rozsah projektu a funkční i nefunkční požadavky.

1.2 Úvod do problematiky

Projekt *Sense* (viz sekce reference) slouží k určování smysluplnosti českých vět na základě předem definovaných morfologických a syntaktických podmínek. Tyto jsou uloženy v textovém souboru ve speciálním formátu, který je součástí dokumentace projektu *Sense*. Pro běžného uživatele je složité a zbytečné se tento formát učit, proto je vhodné poskytnout podpůrnou aplikaci, která tento soubor dokáže spravovat a odstínit uživatele od vnitřního formalismu pomocí přehledného grafického rozhraní. Toto je hlavním úkolem aplikace ConEdit.

1.3 Popis modelu

Základní jednotkou, se kterou bude aplikace pracovat, bude *soubor podmínek*. Jedná se o textový soubor ve formátu UTF-8 obsahující libovolné množství *podmínek*. V praxi je doporučována velikost tohoto souboru do 3000 podmínek, při větším počtu ztrácí aplikace *Sense* na efektivitě. Aplikace *ConEdit* vždy pracuje nejvýše s jedním souborem podmínek, před otevřením jiného je třeba stávající zavřít.

Podmínka je textová definice určitého morfologického vztahu mezi slovy (přesný formát viz sekce 1.4). Každá podmínka tvoří souvislý úsek neprázdných řádků a skládá se ze čtyř částí (právě v tomto pořadí): *komentář* (0-n řádků), *deklarace* (1 řádek), *tělo* (0-n řádků) a *ukončení podmínky* (1 řádek). Mezi podmínkami může být libovolný počet prázdných řádků (0-n).

Komentář: Řádky předcházející podmínce, každý z nich uvozen znaky „/““. Obsah je libovolný.

Deklarace podmínky určuje typ podmínky a slovní druhy, jichž se podmínka týká.

Tělo podmínky obsahuje 0-n *klauzulí*, tedy vlastní obsah podmínky.

Každá *klauzule* je na jednom řádku a popisuje nějaký elementární požadavek na morfologické / syntaktické vlastnosti daných slov. Přesný formát klauzulí viz sekce 1.4.

Ukončení podmínky je jediný řádek obsahující řetězec „end“. Tento řádek končí příslušnou podmínku.

Projekt *Sense* rozeznává tři typy podmínek:

- 1) *Základní podmínky pro smysluplnost dvou slov (1)* – deklarace je uvozena řetězcem *rel*
- 2) *Ternární podmínky pro smysluplnost dvojice slov a spojky / čárky mezi nimi (2)* – deklarace uvozena řetězcem *con*
- 3) *Podmínky pro omezení počtu relací mezi slovy (3)* – deklarace uvozena řetězcem *max*. Tento typ podmínky nemá tělo.

Bližší popis jednotlivých typů podmínek není pro aplikaci ConEdit podstatný a v případě zájmu je součástí dokumentace projektu *Sense* (viz sekce Reference). Zde následuje jen popis formátu jednotlivých podmínek v souboru.

1.4 Stručný popis formátu souboru podmínek

Základní podmínky (Typ (1))

```
rel          slovní druh 1  slovní druh 2
[[!]]1      [hodnota]      [hodnota] ]
[[!]]2      [hodnota]      [hodnota] ]
[[!]]3      [hodnota]      [hodnota] ]
[[!]]4      [hodnota]      [hodnota] ]
...
[[!]]15     [hodnota]      [hodnota] ]
[[!]]lemma  [hodnota]      [hodnota] ]
[[!]]ord    1 / 2          1 / 2      ]
end
```

Ternární podmínky pro spojky (Typ (2))

```
con          slovní druh 1  slovní druh 2
[[!]]1      [hodnota]      [hodnota] ]
[[!]]2      [hodnota]      [hodnota] ]
[[!]]3      [hodnota]      [hodnota] ]
[[!]]4      [hodnota]      [hodnota] ]
...
[[!]]15     [hodnota]      [hodnota] ]
[[!]]lemma  [hodnota]      [hodnota] ]
[[!]]ord    1 / 2          [1 / 2]    ]
[[!]]tok    [hodnota]      ]
end
```

Podmínky pro omezení počtu relací (Typ (3))

```
max          slovní druh 1  slovní druh 2  přirozené číslo
end
```

Deklarace podmínky začíná slovem `rel`, `con` nebo `max` dle typu podmínky. Následují identifikátory dvojice slovních druhů, jíž se podmínka týká. Možné hodnoty zachycuje následující tabulka.

Identifikátory slovních druhů:

<i>Podstatné jméno</i>	N	<i>Příslovce</i>	D
<i>Přídavné jméno</i>	A	<i>Předložka</i>	R
<i>Zájmeno</i>	P	<i>Spojka</i>	J
<i>Číslovka</i>	C	<i>Částice</i>	T
<i>Sloveso</i>	V	<i>Citoslovce</i>	I

Definice podmínky končí slovem `end`. Tělo podmínky tvoří jednotlivé morfologické a syntaktické klauzule, každá na zvláštním řádku. Na pořadí klauzulí nezáleží. V popisu výše jsou nepovinné části uzavřeny mezi symboly `[a]`, všechny klauzule jsou tedy nepovinné a je možné vytvořit i prázdnou podmínku (pouze deklarace a slovo `end`). Volitelně také může každé klauzuli předcházet symbol `!` a každé podmínce komentář na libovolný počet řádků, kde každý je uvozen znaky `//`. Mezi podmínkou a jejím komentářem ani uvnitř komentáře nesmí být žádný prázdný řádek.

Klauzule má dvě až tři části oddělené znakem tabulátoru. První částí je identifikace klauzule (čísla 1-15, `ord`, `tok`, `lemma`), další části jsou příslušné hodnoty pro oba slovní druhy.

Hodnota může být buď jeden konkrétní symbol , rozsah symbolů ve tvaru (a b c) – tedy v závorkách oddělené mezerami, nebo znak + či -. Výčet možných symbolů pro klauzule uvozené čísly 1-15 lze nalézt na adrese <http://ufal.mff.cuni.cz/~hladka/rp200809/cz-appendix-D.pdf>.

Hodnota klauzulí lemma a tok je libovolný řetězec bez mezer, nebo rozsah řetězců opět ve tvaru (str1 str2 str3), nebo znak + či -.

Zápis A / B / C v popisu formátu znamená výčet možných hodnot, tedy po klauzuli ord následuje buď symbol 1, nebo symbol 2.

Podmínka uvozená slovem max je pouze na jeden řádek a obsahuje tabulátorem oddělené identifikátory slovních druhů a přirozené číslo.

1.5 Příklad podmínky v souboru

Takto může vypadat jedna z podmínek uložených v souboru, který má aplikace spravovat:

```
rel          N      R
5            3
ord          2      1
lemma       (k proti naproti oproti kvůli díky)
end
```

Celý soubor potom sestává ze seznamu za sebou jdoucích podmínek podobných této oddělených libovolným počtem prázdných řádků.

1.6 Reference

ConEdit je podpůrnou aplikací pro projekt Sense, jehož dokumentaci je možné nalézt na adrese http://ufal.mff.cuni.cz/~hladka/rp200809/Dokumentace_V_Rovensky.pdf. Dokument obsahuje mimo jiné i přesný formát souboru podmínek, jejich bližší popis a dělení na typy, podle nichž ConEdit podmínky filtruje. Výčet možných hodnot pro všechny morfologické kategorie (klauzule uvozené čísly 1-15) lze nalézt na <http://ufal.mff.cuni.cz/~hladka/rp200809/cz-appendix-D.pdf>.

2. Celkový popis

2.1 Popis aplikace

Aplikace ConEdit musí umět soubor podmínek korektně načíst, zobrazit v grafickém rozhraní a umožnit filtrování, třídění, přidávání, mazání i editaci podmínek bez jakékoli znalosti vnitřního formátu souboru. Rovněž bude umět odstraňovat redundantní podmínky a aktualizovat stav souboru po internetu. Všechny tyto funkční požadavky budou podrobněji popsány dále.

2.2 Běhové prostředí

Aplikace ConEdit bude fungovat pod OS Windows 2000, Windows Xp, Windows Vista, Linux a Solaris, za předpokladu nainstalované Java Virtual Machine verze 6. Dodavatel nezaručuje korektní chování s jinými verzemi JVM / jinými OS. Pro online aktualizace bude vyžadovat přístup k internetu, ostatní funkce budou

dostupné i bez něj. Aplikace bude podporovat přenos souboru podmínek mezi různými podporovanými OS, tedy například soubor podmínek vytvořený v prostředí Windows dokáže aplikace korektně zpracovat i v prostředí Linux.

2.3 Dokumentace

Součástí projektu bude následující dokumentace:

- Uživatelská dokumentace v češtině ve formátu *.pdf* dostupná přímo z aplikace.
- Kontextové nápovědy v aplikaci, které se budou automaticky zobrazovat u kontrolních prvků
- Programátorská dokumentace v angličtině ve formátu *.pdf* popisující architekturu a design programu
- Reference zdrojového kódu v angličtině ve formátu *.html* vygenerovaná programem JavaDoc

2.4 Negativní vymezení

Aplikace bude hlídat korektnost zadávaných dat pouze vzhledem k formátu souboru, nikoli vzhledem k pravidlům konkrétního jazyka, tedy například bude možné zadat klauzuli požadující slovo v osmém pádě atp.

Aplikace bude předpokládat na vstupu korektní soubor podmínek vzhledem ke specifikovanému formátu, v opačném případě bude soubor odmítnut – aplikace nebude obsahovat žádnou možnost opravy poškozeného souboru.

Aplikace je koncipována jako jednouživatelská, není zaručeno korektní chování v případě, že jeden soubor podmínek v jednu chvíli edituje více lidí. Obecně se předpokládá, že v jednu chvíli poběží na počítači jen jedna instance aplikace.

Aplikace nebude rozlišovat mezi různými uživateli.

Aplikace nebude žádným způsobem chránit spravovaná data před přečtením třetí stranou.

Aplikace nebude umožňovat rozšiřování pomocí pluginů, skriptování atp.

3. Funkce aplikace

3.1 Načtení souboru podmínek

Uživatel bude mít možnost ve standardním dialogu (soubor – otevřít) vybrat soubor s podmínkami, který aplikace kompletně načte a zobrazí v přehledné tabulce: v levé části budou náhledy podmínek (dvojice slovních druhů, které se podmínka týká), v pravé části se po vybrání jedné konkrétní podmínky zobrazí její kompletní obsah ve formátu čitelném běžnému uživateli, tj. zkratky používané v interním formátu souboru budou nahrazeny celým názvem příslušné morfologické / syntaktické kategorie.

Po načtení souboru podmínek bude možné je vyfiltrovat podle jejich typu kliknutím na příslušný checkbox (jeden pro každý typ).

Rovněž bude možné nechat soubor podmínek setřídít podle náhledu podmínky tak, aby uživatel viděl podmínky týkající se stejných slovních druhů u sebe. Sémantika třídění bude následující: primárně se třídí dle typu podmínky: nejprve základní podmínky (typ 1), poté ternární podmínky pro spojky (typ 2), nakonec podmínky pro maximální počet relací (typ 3). V rámci jednoho typu podmínek nastupuje sekundární třídění – podle prvního slovního druhu, kterého se daná podmínka týká. V případě dvou podmínek stejného typu se stejným prvním slovním druhem nastupuje terciální třídění – podle druhého slovního druhu, kterého se podmínka týká. Pořadí slovních druhů nechť je standardní: 1 – podstatné jméno, 2 – přídavné jméno, 3 – zájmeno, 4 – číslovka, 5 – sloveso, 6 – příslovce, 7 – předložka, 8 – spojka, 9 – částice, 10 – citoslovce.

3.2 Přidávání, odebírání a editace podmínek

Aplikace bude uživateli umožňovat správu souboru podmínek: na hlavní obrazovce se seznamem podmínek načtených ze souboru bude možné po vybrání podmínky tuto smazat (aplikace zobrazí potvrzovací dialog) nebo upravit, případně vytvořit novou podmínku. Pro vytváření a editaci podmínek bude využit stejný formulář, pouze při přidávání bude zobrazen prázdný, kdežto při editaci podmínek do něj budou automaticky načtena data z editované podmínky. Uživatel tedy pracuje se stejným grafickým rozhraním při editaci i při přidávání.

Přidávací / editovací formulář bude umožňovat zadat vše, co umožňuje formát souboru podmínek, včetně komentářů k jednotlivým podmínkám. To znamená:

Pro podmínky typu (1) slovní druhy, jíž se podmínka týká, výčet klauzulí tvořících tělo podmínky a komentář k podmínce.

Pro podmínky typu (2) navíc klauzule specifické pro tento typ podmínek (tj. klauzule *tok*).

Pro podmínky typu (3) pouze slovní druhy, jichž se podmínka týká, požadované maximum (číslo ≥ 0) a komentář.

Obsah všech zadávaných položek bude aplikace kontrolovat vzhledem k formátu souboru podmínek, při zadání neplatné hodnoty zobrazí chybu a odmítne podmínku uložit.

3.3 Online aktualizace souboru podmínek

Aplikace bude umožňovat aktualizaci souboru podmínek vůči jinému souboru v internetu. Aktualizací se rozumí buď kompletní nahrazení lokálního souboru vzdáleným souborem, nebo doplnění lokálního souboru o ty podmínky ze vzdáleného souboru, které se v lokálním nenacházejí. Pro potřeby aktualizace jsou dvě podmínky považovány za identické, pokud jsou stejného typu a obsahují identické klauzule (na pořadí klauzulí ani na komentáři nezáleží). Identita klauzulí je definována jako rovnost řetězců, tj. dvě klauzule jsou identické, právě když jim příslušné řádky v souboru jsou stejné až na mezery a tabulátory.

Uživatel bude moci nastavit *URI* souboru, vůči kterému má aktualizace proběhnout.

Po dokončení aktualizace bude uživatel informován o výsledku, tj. zda proběhla úspěšně a pokud ano, jaký byl počet stažených podmínek a na přání uživatele zobrazí kompletní výpis přidaných podmínek.

Pro přenos bude využíván protokol HTTP 1.1 (standardními mechanismy jazyka Java – např. třída `java.net.HttpURLConnection`).

Pro korektní běh aktualizace bude potřeba přístup k internetu (stabilní spojení bez výpadků). Vzdálený soubor musí být ve formátu, který produkuje aplikace ConEdit. Defaultně bude komunikace probíhat na portu 80, půjde ale nastavit i jiný. Na tomto portu na cílovém stroji musí běžet HTTP server poskytující daný aktualizací soubor (stačí pouze pro čtení). Aplikace nezaručuje korektní chování aktualizace v případě konfliktu se SW třetích stran (např. firewallly blokující spojení, antiviry).

Při jakémkoli selhání spojení aplikace zahlásí chybu a její příčinu a zanechá lokální soubor beze změn. To platí i v případě, že se spojení vůbec nepodaří navázat.

3.4 Odstraňování redundantních podmínek

Jakmile se v souboru vyskytne větší množství podmínek, je určitá pravděpodobnost, že některé budou redundantní. Podmínka A je redundantní, pokud někde v souboru existuje podmínka B, která je méně přísná, tj. její klauzule jsou podmnožinou klauzulí podmínky A. Aplikace bude k zjišťování této skutečnosti využívat jednoduchý algoritmus: nechť BÚNO A je podmínka s menším počtem klauzulí než B. Pak pro každou klauzuli v těle podmínky A hledáme identickou klauzuli v těle podmínky B (projdeme všechny klauzule v těle podmínky B). Identita klauzulí je definována v sekci 3.3.

Popis algoritmu:

Pro každou klauzuli K1 v těle podmínky A

{

 Pro každou klauzuli K2 v těle podmínky B

 Pokud jsou K1 a K2 identické, pokračuj na další klauzuli z A

 Pokud neexistuje klauzule z B identická K1, vrať false

}

Vrať true

Je zřejmé, že vždy když uspěje podmínka A, uspěje i B (ne naopak) a podmínka A je proto redundantní.

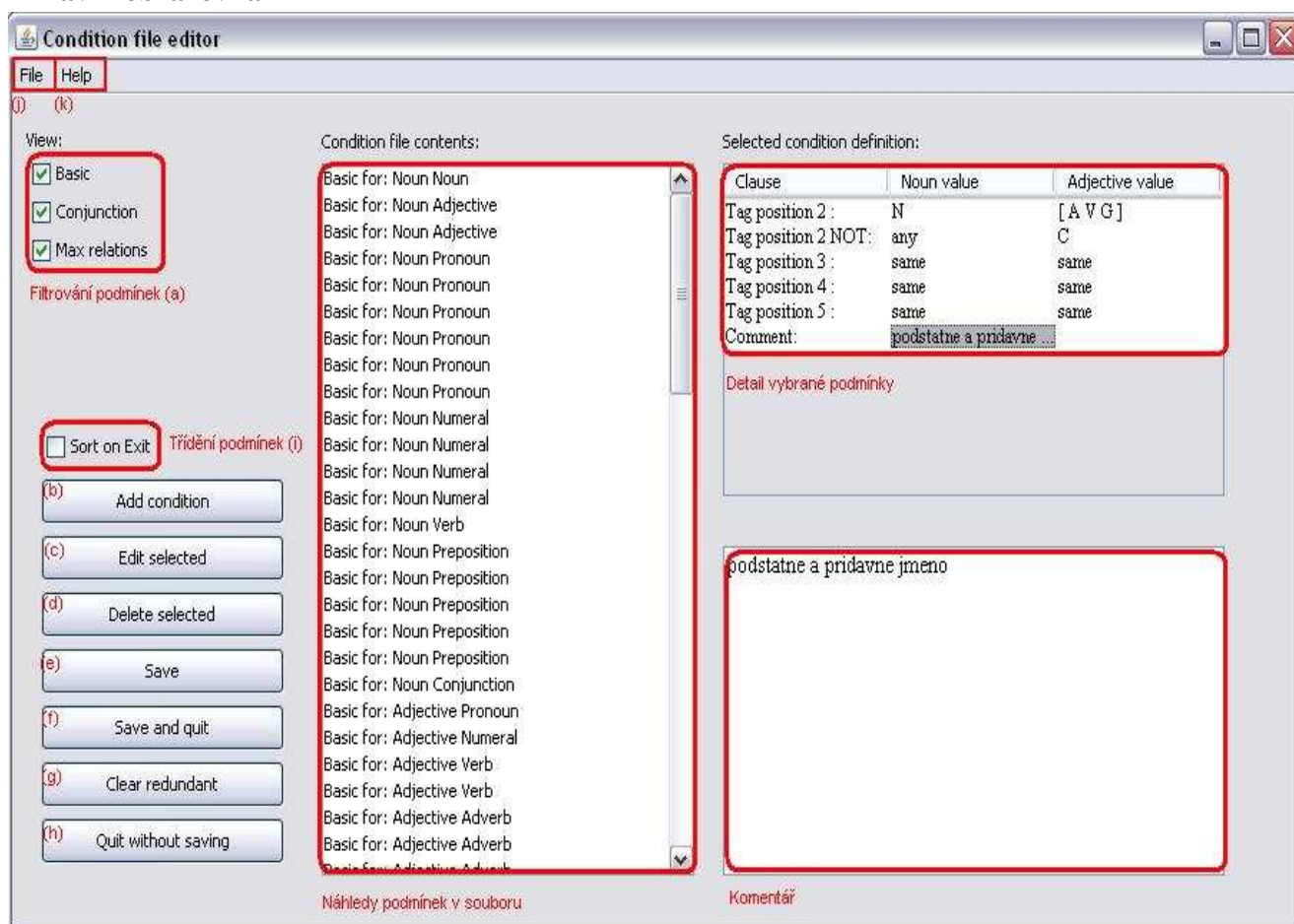
Aplikace bude umožňovat automatické vyhledávání a odstraňování těchto redundantních podmínek.

Po skončení tohoto procesu zobrazí aplikace počet odstraněných podmínek a na přání uživatele také jejich kompletní výpis.

4. GUI

GUI aplikace se bude skládat ze dvou základních obrazovek: *hlavní obrazovka* poskytne uživateli veškerou funkčnost týkající se správy souboru podmínek a *přídávací/editovací formulář*, ve kterém bude možné přidávat i editovat podmínky – pro obojí bude stejné GUI, při editaci bude formulář předvyplněn daty z editované podmínky.

Hlavní obrazovka



- a) V poli „Condition file contents“ se zobrazují jen podmínky zde zaškrtnutých typů
- b) Otevře obrazovku pro přidávání podmínek (viz níže)
- c) Jako b) + obsah otevřeného formuláře předvyplní klauzulemi podmínky vybrané v poli „Condition file contents“
- d) Vymaže právě vybranou podmínku
- e) Uloží veškeré změny do souboru podmínek
- f) Jako e) + ukončí aplikaci
- g) Spustí proces odstraňování přebytečných podmínek
- h) Ukončení aplikace bez provedení změn
- i) Při zaškrtnutí této volby bude soubor podmínek těsně před ukončením aplikace seříděn
- j) Obsahuje volby Open – zobrazí standardní dialog pro výběr souboru podmínek, Exit – Ukončí aplikaci
- k) Zobrazí nápovědu

Přidávání podmínek typu (1) a (2)

Basic Condition Conjunction Condition Max Relations condition

Declaration

First token part of speech: Noun

Second token part of speech: Noun

Výběr slovních druhů

Definition Přidávání klauzulí podmínky

Tag Position #2: [v] Negated: []

First token: [Any] [Same] [Value]

Second token: [Any] [Same] [Value]

Add this line(a) Remove selected(b)

[x] Word order

[x] This one first [] This one first

Definice slovosledu

Lines found

Zde se zobrazí již zadané klauzule

Comment: Example condition|

Save condition (c) Reset(d) Storno (e)

- Přidá novou klauzuli do podmínky, zobrazí se v poli „Lines found“
- Odstraní klauzuli vybranou v poli „Lines found“
- Uloží vybranou podmínku, ne však přímo do souboru podmínek, jen do pomocných struktur, uložení změn do souboru spouští uživatel explicitně na hlavní obrazovce
- Vyprázdní všechna editovací pole na této obrazovce
- Zavře tuto obrazovku

Přidávání podmínek typu (3)



The screenshot shows a software window with a title bar and standard Windows window controls (minimize, maximize, close). The window contains three tabs: 'Basic Condition', 'Conjunction Condition', and 'Max Relations condition'. The 'Max Relations condition' tab is currently selected and highlighted. Below the tabs is a section titled 'Declaration & Definition' which contains three input fields: 'First token part of speech' with a dropdown menu showing 'Noun', 'Second token part of speech' with a dropdown menu showing 'Noun', and 'Max number of relations' with a text input field containing the number '5'. Below this section is a large, empty rectangular area. At the bottom of the window is a 'Comment' section with a text area containing the text 'Example condition'. To the right of the text area are vertical scroll arrows. At the very bottom of the window are three buttons: 'Save Condition', 'Reset', and 'Storno'.

Basic Condition Conjunction Condition Max Relations condition

Declaration & Definition

First token part of speech Noun

Second token part of speech Noun

Max number of relations 5

Comment

Example condition

Save Condition Reset Storno

5. Nefunkční požadavky

5.1 Lokalizace

Aplikace bude kompletně v češtině, včetně uživatelské dokumentace a kontextových nápověd. Programátorská dokumentace bude v angličtině.

5.2 Minimální HW a SW požadavky

Aplikace bude ke spuštění vyžadovat PC s CPU o frekvenci alespoň 1Ghz, 256 MB RAM operační paměti a prostor na disku 10 MB. Rovněž bude vyžadovat nainstalovanou JVM verze 6, OS Windows 2000, Windows XP, Windows Vista, Linux nebo Solaris.

5.3 Výkon

Veškeré požadavky na výkon předpokládají standardní výkon ke dni psaní specifikace tj. dvoujádrový procesor 2.5 Ghz, 1GB RAM, nesdílené připojení k internetu rychlosti 512Kbit/s. Za předpokladu, že na počítači nepoběží žádné výpočetně náročné procesy a soubor podmínek nepřesáhne doporučovanou maximální velikost 3000 podmínek bude splněno následující:

- načtení souboru podmínek nebude trvat déle než 3 vteřiny
- Třídění podmínek nebude trvat déle než 3 vteřiny
- Aktualizace souboru podmínek nebude trvat déle než 5 vteřin
- Ostatní operace (mazání/ukládání/zobrazení podmínky) nebudou trvat déle než 1 vteřinu

5.4 Další požadavky

Dostupnost

Funkce on-line aktualizace bude dostupná jen v případě funkčního připojení (viz 3.3). Ostatní funkce aplikace budou dostupné i bez připojení k internetu.

Robustnost

V případě, že dojde k nekorektnímu ukončení programu z jakéhokoli důvodu (výpadek proudu, zahlcení paměti, výpadek sítě...) a v jakoukoli chvíli, zanechá aplikace soubor podmínek beze změny, avšak v korektním stavu – to bude realizováno explicitním ukládáním změn. Veškeré změny budou prováděny pouze nad pomocnou kopií souboru, uložení změn do vlastního souboru podmínek iniciuje uživatel.

Aplikace zaručuje korektnost souboru podmínek v případě úprav v rámci aplikace, nezaručuje však korektnost souboru ani korektní chování aplikace v případě jakýchkoli úprav souboru mimo aplikaci (například přímá úprava v některém textovém editoru).

Aplikace bude kontrolovat veškerý vstup od uživatele vzhledem k formátu souboru podmínek a nepovolí žádnou změnu, která by tento formát porušovala.

Zabezpečení

Aplikace nebude žádným způsobem zajišťovat utajení dat, autentizaci ani autorizaci uživatelů. Veškerá data budou ukládána v plain-textu nešifrovaná, dostupná komukoli a to i při on-line aktualizaci.

Rozšiřitelnost

Aplikace nemusí být psána s důrazem na rozšiřitelnost, množina funkčních požadavků je konečná, nemusí obsahovat žádný systém SPI (pluginů).

Převoditelnost

Aplikace nemusí být spustitelná ani převoditelná na žádný OS, který není explicitně uveden v SW požadavcích aplikace.

Škálovatelnost

Aplikaci není nutné optimalizovat pro velké objemy dat, funkcionality musí být zajištěna (a otestována) pro velikosti souboru cca 3000 podmínek.

Znovupoužitelnost

Jediná část kódu určená k opětovnému použití bude parsování souboru podmínek – tato část bude podrobně zdokumentována a nezávislá na zbytku aplikace tak, aby bylo možné ji zakomponovat do jiné aplikace. Bude se jednat o knihovnu umožňující obsah souboru podmínek automaticky narpasovat do interních struktur a poskytnout jej uživateli této knihovny prostřednictvím veřejného API knihovny. Návrh této knihovny bude popsán v UML schématu, které bude součástí dodávky.

Testovatelnost

Součástí dodávky bude sada testů pokrývajících kompletní veřejné rozhraní parseru v poměru nejméně jeden test na jednu veřejnou metodu. Rozhraní parseru necht' je psáno s důrazem na modulárnost pro usnadnění případného dalšího testování. Testy pro zbytek aplikace nemusí být součástí dodávky.

Použitelnost

Aplikace bude vyvíjena s důrazem na snadné použití. Cílem je intuitivní práce s minimální potřebou nápovědy / dokumentace. Pro ověření uživatelské přívětivosti bude GUI aplikace před dodávkou otestováno a schváleno alespoň jedním běžným uživatelem. V zájmu zlepšení použitelnosti je možné provést úpravy v uvedených schématech GUI.

Správa aplikace

Správa aplikace by měla být téměř nulová. Aplikace si nebude vytvářet žádné dlouhodobé soubory s výjimkou souboru podmínek, veškeré dočasné soubory bude automaticky mazat, nebude provádět žádné změny v registrech ani konfiguraci systému. S aplikací bude dodáván automaticky instalátor i odinstalátor, který ji ze systému kompletně odstraní (až na soubory podmínek). Oba nástroje budou v podobě hotového skriptu v některém standardním jazyce (shell / batch), nebo coby samoinstalační balíček (např. .msi soubor pro Windows).

Příloha A: Používané pojmy

URI – Uniform Resource Identifier, jednoznačné umístění vzdáleného souboru, viz <http://www.ietf.org/rfc/rfc2396.txt>