

UNIVERZITA KARLOVA V PRAZE

---

Matematicko-fyzikální fakulta

# DIPLOMOVÁ PRÁCE

Petr Václavek  
Klient DIS

Katedra softwarového inženýrství  
Vedoucí diplomové práce: Mgr. Michal Kopecký  
Studijní program: Informatika

---

listopad 2001

Rád bych poděkoval panu magistru Michalovi Kopeckému za vedení a podporu při tvorbě diplomové práce, Slávkovi Rydvalovi za pomoc při sazbě a Honzovi Palovi za vyčerpávající korekturu.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 10. prosince 2001

Petr Václavek

# Obsah

<b>1</b>	<b>Úvod</b>	<b>7</b>
<b>2</b>	<b>Specifikace diplomové práce</b>	<b>9</b>
<b>3</b>	<b>Vyhledávání na Internetu</b>	<b>10</b>
3.1	Katalogové vyhledávače . . . . .	10
3.2	Fulltextové vyhledávače . . . . .	11
3.2.1	Roboti . . . . .	12
3.2.2	Databáze . . . . .	12
3.2.3	Uživatelské prostředí (možnosti vyhledávacího stroje) . . . . .	13
3.2.4	Metavyhledávače . . . . .	15
3.2.5	Klientské vyhledávače . . . . .	15
<b>4</b>	<b>Obdobné aplikace</b>	<b>17</b>
4.1	Copernic 2001 . . . . .	17
4.2	Bulls Eye 2 . . . . .	20
4.3	Babylon . . . . .	22
<b>5</b>	<b>Programátorská dokumentace (implementace)</b>	<b>25</b>
5.1	Vývoj architektury . . . . .	25
5.2	Popis rozhraní . . . . .	28
5.2.1	Systém parametrů . . . . .	28
5.2.1.1	TDISParam . . . . .	31
5.2.1.2	TDISScalarParam . . . . .	31
5.2.1.3	TDISGroupParam . . . . .	32
5.2.1.4	Implementace parametrů . . . . .	32
5.2.1.5	Pole parametrů . . . . .	33
5.2.1.6	Příklady parametrů . . . . .	34
5.2.2	Server . . . . .	35
5.2.2.1	Struktura DLL knihovny . . . . .	35
5.2.2.2	Vlastní server . . . . .	36
5.3	Průběh položení dotazu . . . . .	36
5.4	Aplikace . . . . .	40
5.4.1	Výstupní šablony . . . . .	40
5.4.1.1	Popis souboru šablony . . . . .	42
5.4.2	INI soubory . . . . .	42
5.5	Implementace jednotlivých serverů . . . . .	44
5.5.1	Implementace webových vyhledávačů . . . . .	44

---

5.5.1.1	Parser . . . . .	46
5.5.2	Implementace Oracle8i interMedia . . . . .	50
<b>6</b>	<b>Závěr</b> . . . . .	<b>51</b>
6.1	Srovnání s konkurenčními produkty . . . . .	51
6.2	Další vývoj . . . . .	52
6.3	Zhodnocení a závěr . . . . .	52
<b>A</b>	<b>Uživatelská dokumentace</b> . . . . .	<b>55</b>
A.1	Předmluva . . . . .	55
A.2	Požadavky . . . . .	55
A.2.1	Minimální konfigurace . . . . .	55
A.2.2	Doporučená konfigurace . . . . .	57
A.2.3	Poznámky ke starším operačním systémům . . . . .	57
A.3	Instalace . . . . .	57
A.4	Práce s aplikací . . . . .	59
A.4.1	Popis aplikace . . . . .	59
A.4.2	Přihlášení . . . . .	61
A.4.3	Vytvoření a editace dotazu . . . . .	61
A.4.4	Vyhledávání . . . . .	64
A.4.5	Nastavení . . . . .	67
A.4.6	Seznamy dokumentů, export . . . . .	68
A.4.7	Další možnosti aplikace . . . . .	69
A.4.8	Využívané servery . . . . .	71
A.5	Vlastní šablona pro export . . . . .	71
A.5.1	Popis souboru šablony . . . . .	71
A.6	Klávesové zkratky . . . . .	73
<b>B</b>	<b>Obsah příloženého CD</b> . . . . .	<b>74</b>

# Seznam tabulek

4.1	Informace o aplikaci Copernic 2001 . . . . .	19
4.2	Informace o aplikaci Bulls Eye 2 . . . . .	22
4.3	Informace o aplikaci Babylon . . . . .	24
5.1	Názvy a popisy tagů šablony . . . . .	43
A.1	Názvy a popisy tagů šablony . . . . .	72
A.2	Klávesové zkratky . . . . .	73

# Seznam obrázků

4.1	Copernic . . . . .	18
4.2	Bulls Eye . . . . .	20
4.3	Babylon . . . . .	23
5.1	Výchozí architektura aplikace . . . . .	26
5.2	Finální architektura aplikace . . . . .	27
5.3	Prapůvodní popis rozhraní . . . . .	28
5.4	Objekty rozhraní . . . . .	29
5.5	Objekty rozhraní . . . . .	30
5.6	Využití balíčku DISifacePackage . . . . .	33
5.7	Objekty jádra aplikace (DISEngine) . . . . .	37
5.8	Sestavení pole parametrů dotazu . . . . .	38
5.9	Strom parametrů . . . . .	41
5.10	Implementace serverů . . . . .	45
5.11	Grafické znázornění prvků tokenu v textu . . . . .	47
A.1	Schéma práce aplikace DISClient . . . . .	56
A.2	Úvodní dialog . . . . .	57
A.3	Výběr adresáře . . . . .	58
A.4	Instalace připravena . . . . .	58
A.5	Instalace úspěšně dokončena . . . . .	59
A.6	Hlavní okno aplikace . . . . .	60
A.7	Dialog pro přihlášení . . . . .	61
A.8	Základní vlastnosti dotazu . . . . .	62
A.9	Výběr serverů . . . . .	62
A.10	Parametry dotazu . . . . .	63
A.11	Nalezené chyby dotazu . . . . .	64
A.12	Průběh vyhledávání . . . . .	65
A.13	Seznam nalezených dokumentů . . . . .	65
A.14	Chyby vzniklé při vyhledávání . . . . .	66
A.15	Statistika vyhledávání . . . . .	66
A.16	Nastavení . . . . .	67
A.17	Nastavení serverů . . . . .	68
A.18	Export seznamu dokumentů . . . . .	69
A.19	Přehled všech parametrů . . . . .	70
A.20	Seznamy dotazů . . . . .	70
A.21	Seznam dostupných serverů . . . . .	71

# Kapitola 1

## Úvod

Svět je v současnosti zahlcován velkým množstvím informací (tiskové zprávy, výpisy, korespondence, příspěvky elektronických konferencí, odborné články a v poslední době už i celé knihy), které je nutno zpracovat a uložit k pozdějšímu použití. Toto zpracování a uložení se naštěstí již převážně provádí v elektronické podobě. Pod pojmem pozdějšího použití se rozumí vyhledávání důležitých informací v těchto archívech.

Bohužel neexistuje jednotný formát těchto dat, natož pak jednotné zpracování, archivování a nástroje pro vyhledávání. Existuje nepřehledné množství způsobů (formátů), jak tyto data uložit a další nové formáty vznikají. Tím se hledání v textech stává komplikovanější, protože na každý formát dat existují jiné nástroje. Některé formáty dokonce žádný rozumný způsob prohledávání neumožňují.

Mezi nejčastěji používané zdroje textových dat, se kterými se běžný uživatel počítače setká patří následující.

**E-mailová korespondence a příspěvky elektronických konferencí.** Texty dopisů jsou uloženy v poštovních klientech (Microsoft Outlook, PegasusMail, Pine, Lotus Notes a další). Data v těchto aplikacích jsou ukládána ve vlastním formátu, který je pro další využití ostatními aplikacemi většinou naprosto nevhodný. Naštěstí většina těchto poštovních klientů umožňuje exportovat data do čistého textu, se kterým se pracuje podstatně lépe.

**Samostatné dokumenty na disku v různých formátech.** K většině níže uvedeným formátům existují aplikace pro vyhledávání. Téměř vždy to jsou aplikace, ve kterých zmíněné typy dokumentů vznikají. Tyto nástroje jsou omezené pouze na daný formát.

Existuje i několik výjimek, například Microsoft Index Server (součást Internet Information Server a Windows NT Serveru, sloužící pro indexaci a plnohodnotné vyhledávání dokumentů přístupných přes WWW) pracuje s formáty txt, html a s dokumenty kancelářského balíku MS Office.

- Čistý text (.txt) – platformě nezávislý, snadno přenositelný a nestrukturovaný (moc se nepoužívá).
- Elektronická kniha (.lit), neboli e-Book, prohlížitelná pomocí Microsoft Reader (ovšem obdobné formáty vyvíjí i jiné firmy, jako například Adobe).
- Dokumenty Microsoft Office (.doc). Velmi nepřenositelné, dokonce se jedná o formát nekompatibilní mezi jednotlivými verzemi Office. Bohužel je velmi

---

oblíben, i když podstatně lepší vlastnosti (co do zpracování a přenositelnosti) má Rich Text Format (.rtf). Tento formát se celkem dost rozšířil a většina textových procesorů s tímto formátem dokáže pracovat (importovat a exportovat data).

- Portable Document Format (.pdf) – platformě nezávislý formát firmy Adobe, který lze prohlížet například pomocí Adobe Acrobat Reader.
- PostScript (.ps) – textový soubor popisující vzhled tištěné stránky včetně textu a obrázků. Je podporován nejen výrobci softwaru, ale i hardwaru. Tyto soubory lze v prostředí Windows prohlížet například pomocí Ghost View.
- Webové stránky (.htm, .html, .shtml, ...) jsou psány jazykem HTML (Hypertext Markup Language). Téměř každý operační systém obsahuje prohlížeč tohoto formátu.
- XML (.xml) – univerzální a otevřený formát (Extensible Markup Language) pro výměnu dat. V poslední době velmi oblíbený.

**Informace v databázích.** Velké množství textů je uloženo v databázích. Je to jeden z nejlepších způsobů pro uložení dat. Nabízí poměrně snadnou správu dat, umožňuje vyhledávání pomocí SQL příkazů.

V současné době většina databázových serverů obsahuje buď speciální rozšíření pro ukládání textů (například u serveru Oracle bylo dříve rozšíření ConText Cartridge, nyní Oracle8i interMedia Text) případně alespoň nabízí operátory umožňující rozumně vyhledávat (například v Microsoft SQL Serveru jsou operátory CONTAINS, CONTAINSTABLE, FREETEXT, FREETEXTTABLE).

**Internet.** Ovšem nejvíce textů se nachází na Internetu v podobě webových stránek. I v tomto případě existuje způsob pro vyhledávání. Jsou jím různé webové služby (katalogové a fulltextové vyhledávače, hledání v konferencích, ...).

Bohužel v současné době neexistuje žádný způsob, jak ve výše uvedených zdrojích informací hromadně vyhledávat. Neexistuje dokonce ani definice jazyka, který by byl jednotný pro specifikaci dotazu. Každá aplikace si proto vytváří vlastní jazyk pro popsání hledaných termů a vztahů mezi nimi.

V databázových systémech se používá jazyk SQL (Structured Query Language) rozšířený o operátory pro fulltextové hledání, webové vyhledávače používají syntaxi boolovské algebry rozšířenou o vlastní operátory pro hledání v různých sekcích, doménách, ...

Uživatelé se tak k mnohým důležitým informacím ani nedostanou, protože nejsou schopni (případně jsou již znechuceni) využít a prohledat všechny zdroje dat. A nejsou ani schopni plně využít možnosti, které jednotlivé nástroje nabízí. Většinou se zaměří na několik málo nástrojů, které si více osvojí a tyto používají.



# Kapitola 2

## Specifikace diplomové práce

Cílem diplomové práce je vytvořit program usnadňující vyhledávání informací v dokumentech, jež jsou uloženy na rozličných místech a v různých formátech. Jedná se tedy o program podobný webovému metavyhledávači (viz [Oddíl 3.2.4](#)) nebo spíše klientskému vyhledávači (viz [Oddíl 3.2.5](#)), ovšem s tím rozdílem, že nebude oslovovat pouze webové vyhledávací stroje, ale i další libovolné zdroje dat (Oracle a jiné databáze, soubory na lokálním disku, ...).

Množina zdrojů, které se budou pro vyhledávání používat, půjde snadno rozšířit o další, a to bez nutnosti rekompilace aplikace. Navíc toto rozšíření zdrojů bude umožněno komukoliv – bude implementováno v nezávislém modulu, který bude sloužit pro komunikaci mezi příslušným fyzickým serverem a aplikací.

Hlavní náplní je tedy navrhnout a implementovat rozhraní mezi aplikací a vlastním DIS serverem (který bude oslovován) tak, aby umožnilo uživateli:

- Maximálně využívat možnosti příslušného serveru.
- Nenutilo jej zadávat opakovaně stále ty samé údaje pro každý server, který má být dotazem osloven.
- Používat jednotnou syntaxi v dotazu pro všechny servery.
- Vytvořit další modul plnohodnotně využívající jiné zdroje dat.

Základní funkce a rysy výsledné aplikace:

- Uživatelsky přívětivé a víceuživatelské prostředí.
- Zadávání dotazu pro několik různých serverů najednou.
- Vlastní vyhledávání relevantních dokumentů.
- Uchovávání a správa historie dotazů.
- Zpracování výsledků do jednotné podoby.
- Získání (zobrazení, uložení) nalezených dokumentů.
- Ukládání a načítání dotazů, výsledků, ...
- Export výsledků v několika formátech (HTML, Plain text, a případně další).

# Kapitola 3

## Vyhledávání na Internetu

Vyhledávání informací na Internetu se dá rozdělit do několika kategorií podle použitých nástrojů:

- Katalogové vyhledávače.
- Fulltextové vyhledávače.
- Metavyhledávače.
- Klientské vyhledávače.

### 3.1 Katalogové vyhledávače

Katalogové vyhledávače jsou postaveny na velké databázi stránek, která má stromovou strukturu. Jednotlivé odkazy jsou rozříděny do několika (řádově deseti) základních kategorií, které se dále dělí na podkategorie a tak dále.

Uživatel může tyto služby využívat dvojm způsobem:

- Prochází jednotlivé kategorie, které ho zajímají.
- Použije jednoduché vyhledání v této stromové struktuře.

Odkazy jsou do databáze zadávány ručně (většinou se nepoužívají žádné programy, které samy procházejí Internet), a to buď autory stránek (každý katalog obsahuje formulář pro zařazení stránek do databáze. Vyřízení žádosti trvá několik dní, neboť jsou stránky před vlastním přidáním do seznamu zkontrolovány) nebo vlastními správci katalogu.

Mezi nejznámější katalogy u nás patří:

- Seznam (<http://www.seznam.cz>) – první a nejznámější katalog u nás.
- Atlas (<http://www.atlas.cz>)
- Centrum (<http://www.centrum.cz>)
- Quick (<http://search.quick.cz>)
- RedBox (<http://www.redbox.cz>)

Naprostá většina z nich (jak uvádí [12]) využívá data projektu „DMOZ – open directory project“. Je to největší a nejúplnější katalog na webu, který spravuje komunita dobrovolných uživatelů. Stal se základem většiny katalogových vyhledávačů nejen u nás, ale i ve světě (Netscape Search, AOL Search, Google, Lycos, HotBot, DirectHit, a stovek dalších).

V zahraničí jsou asi nejznámější:

- DMOZ (<http://dmoz.org>) – Open Directory Project.
- Yahoo (<http://www.yahoo.com>)
- Excite (<http://www.excite.com>)
- Lycos (<http://dir.lycos.com>)
- Google (<http://directory.google.com>)

## 3.2 Fulltextové vyhledávače

Na rozdíl od katalogových vyhledávačů se stránky v jejich databázích nedají sekvenčně procházet. Umožňují pouze nalézt relevantní stránky odpovídající zadanému dotazu.

Většina vyhledávačů tohoto typu používá „Boolovský model“. Jedná se o jeden z nejstarších modelů DIS, jehož základy byly navrženy již v 50-tých letech dvacátého století. Každý dokument modelu obsahuje množinu slov a vyhledávání je založeno na vyhodnocování boolovských výrazů (AND, OR, NOT) jak je uvedeno v [1] a [2].

Například dotazem  $Z \text{ AND } (X \text{ OR } Y)$  se naleznou dokumenty obsahující slovo  $Z$  a alespoň jedno ze slov  $X$  nebo  $Y$ .

Tento model lze dále rozšířit o operátory vzájemné polohy termů v dokumentu (například term  $X$  musí být maximálně ve vzdálenosti 10 slov od termu  $Y$ , případně ve stejném odstavci, stránce, kapitole, ...).

Hlavní nevýhodou boolovského modelu je nemožnost ohodnotit slova (určit, která slova jsou důležitá a která nikoliv. Typicky běžná slova, jako například „který“, „kolem“ a podobně, jsou v článku o JavaScriptu podstatně méně důležitá než „toString“, „getDate“, ... ) a to jak v dotazu, tak v samotném dokumentu, což má několik špatných důsledků:

- Obtížná formulace dotazů.
- Nelze ohodnotit nalezené záznamy dle relevance (všechny dokumenty, které odpovídají dotazu jsou stejně dobré).
- Termy v dotazu i dokumentu jsou chápány jako stejně důležité.
- Neintuitivní výsledky.  $(T_1 \text{ OR } T_2 \text{ OR } \dots \text{ OR } T_n)$  – vrátí záznamy se všemi  $T_i$ , ale i takové, které obsahují jedině  $T_i$ .  $T_1 \text{ AND } T_2 \text{ AND } \dots \text{ AND } T_n$  – ve výsledku nebudou záznamy neobsahující jedině  $T_i$ .

Tento nedostatek řeší vektorový model, kde jsou termy hodnoceny vahou (0-1) a dotaz je reprezentován jako množina termů a jejich ohodnocení. Vyhledávání poté spočívá v porovnávání vah termů v dotazu a dokumentu.

Fulltextové vyhledávací stroje se skládají z několika částí:

- Roboti, kteří procházejí web
- Databáze
- Uživatelské prostředí

### 3.2.1 Roboti

Jinak též zvaní „spiders“, „worms“ či „web crawlers“ jsou programy, které procházejí web a indexují nalezené stránky. Pod pojmem Indexace se rozumí zpracování obsahu stránky – tedy zapsání všech důležitých klíčových slov stránky (a dalších informací) do databáze.

Kromě hledání nových stránek mají za úkol prověřovat aktuálnost databáze. To znamená, že kontrolují zda v ní obsažené dokumenty jsou stále ještě přístupné a zda jejich obsah odpovídá uloženým informacím. Během jednoho dne stihnou tyto programy navštívit několik milionů stránek.

Některé fulltextové vyhledávače umožňují návštěvníkům vložit vlastní stránky do databáze (jiné toto neumožňují a spoléhají pouze na své roboty). Samozřejmě, že se při přijetí požadavku neprovede zápis přímo do databáze, ale uvedená adresa se podstrčí robotovi, který ji zpracuje jako ostatní stránky.

Roboti kromě zapsání informací o stránce zjistí také adresy, na které se odkazuje a vloží je do fronty odkazů, které později zpracuje (navštíví a naindexuje). Je tedy zřejmé, že čím více odkazů na stránku ukazuje, tím dříve bude nalezena a zařazena do indexu. Pokud na stránku neexistuje žádný odkaz, nebude nalezena nikdy a pokud nebude zařazena do databáze ručně, může se stát, že nebude nikdy naindexována.

Díky tomu, že se indexování provádí automaticky, existují stránky, které nemohou být nikdy zařazeny do databáze. Jsou to zejména:

- Stránky kde se vyžaduje registrace či heslo.
- Dynamické stránky generované z databází.
- Komentáře ve zdrojových textech stránky.

Občas se může stát, že si autor nepřeje, aby byly jeho stránky zařazeny do databáze (například před oficiálním zveřejněním webu a podobně). Většinou to lze zařídit, díky tomu, že roboti by měli splňovat „Robot Exclusion Standard“ (viz [6]). Není tedy problém zajistit pomocí metatagů ve stránce, případně pomocí jednoho souboru v kořenovém adresáři serveru, vypnutí indexace.

### 3.2.2 Databáze

Obsahuje informace o naindexovaných stránkách. Kromě seznamu klíčových slov (termů), které stránka obsahuje, se můžou ukládat i jiné informace: pořadí slov, zda slova tvoří nějakou známou frázi, datum modifikace stránky, linky z této stránky a na tuto stránku, informace o serveru, kde se stránka nachází, zda je mládeži přístupná, atd. Záleží na implementaci daného systému. Čím více informací se ukládá, tím více možností má uživatel při zadávání dotazu a také při prohlížení výsledků.

### 3.2.3 Uživatelské prostředí (možnosti vyhledávacího stroje)

Základní možnosti, které fulltextové vyhledávače podporují (viz [4] a [5]) jsou:

- Vyhledávání frází (slova, která se v hledaném dokumentu musí vyskytovat u sebe). Při zadávání se používají uvozovky.
- Podpora základních operátorů: AND, OR, NOT, NEAR (spojuje slova, která by se v dokumentu měla vyskytovat blízko sebe). Některé stroje tyto operátory zjednodušují na znaménkový prefix:
  - „+“ dokument musí obsahovat slovo uvedené za tímto znakem.
  - „-“ dokument nesmí obsahovat následující slovo.

Většinou je možné místo těchto operátorů psát jejich zkratky ( &, |, !).

- Podpora žolíků „\*“ slouží pro nahrazení libovolného počtu písmen (někdy je z důvodu optimalizace umožněno použít žolíka pouze v případě, že je zadán jistý počet písmen. Navíc je někdy povoleno používat žolíky pouze jednostranně, to jest že představují jen libovolnou koncovku, případně předponu).
- Volba jazyku stránky (několik desítek možností).
- Datum poslední modifikace.
- Omezení počtu výsledků z jednoho serveru. Stránky na jednom serveru mají většinou velmi podobný obsah (zaměření). Takto lze získat více relevantních stránek z více serverů (z každého serveru se vybere pouze jedna, či několik málo, nejrelevantnějších stránek).
- Slova v různých sekcích stránky (text odkazu, název obrázku, odkaz, adresa, název).
- Zúžení vyhledávání pouze na určitou doménu nebo server.

Velmi důležité u vyhledávacích serverů je také způsob hodnocení dokumentů a tedy jejich uspořádání na výstupu. Pořadí dokumentu může záležet na mnoha faktorech:

- Obsahuje-li hledané termy v názvu stránky (HTML tag <TITLE>).
- Zda jsou hledaná slova uvedena na začátku textu stránky.
- Počet nalezených slov a jejich výskytů v textu (čím více z hledaných slov dokument obsahuje, tím má lepší ohodnocení). Proto je rozumné zadávat v dotazu hodně podobných termů.
- Obsahuje-li více slov, které jsou označeny znaménkem „+“ v dotazu.
- Obsahuje-li slova, která byla nalezena na relativně málo jiných stránkách.
- Zda jsou hledaná slova v dokumentu blíže u sebe.
- Zda dokument obsahuje některá slova jako frázi, ačkoliv tato slova nebyla v dotazu uvedena v uvozovkách (například často hledané spojení typu „Bill Gates“ a podobně).

Později začaly fulltextové vyhledávače nabízet i další rozšiřující služby, které s vyhledáváním souvisí:

- Automatický překlad zobrazovaných stránek do jiných jazyků.
- Hledání podobných stránek.
- Další hledání mezi již nalezenými stránkami (ladění dotazu).
- Vrácení dalších relevantních stránek z vybraného serveru (v případě, že je zapnuta filtrace serverů).

Příklady fulltextových vyhledávačů u nás:

- Kompas (<http://kompas.seznam.cz>) – Součást nejznámějšího českého katalogu Seznam.
- Megatext (<http://www.megatext.cz>) – Chlubí se, že patří mezi největší a technologicky nejpokročilejší fulltextové vyhledávače u nás. A to zejména díky tomu, že se jako jediný z velkých českých vyhledávačů dokáže plně vypořádat s klíčovými problémy složité české gramatiky (viz [10]). Umí vyhledávat česká ohýbaná slova ve všech jejich morfologických tvarech. Má pro řešení tohoto problému k dispozici rozsáhlý slovník 50-ti tisíc nejfrekventovanějších českých slov, s jehož pomocí dokáže hledané výrazy spolehlivě identifikovat bez ohledu na to, v jakém pádu, čísle nebo slovesném čase se daný termín na stránce vyskytuje.
- Atlas (<http://hledej.atlas.cz>)
- Redbox (<http://www.redbox.cz>)

V zahraničí:

- Altavista (<http://www.altavista.com>) – Jeden z prvních a nejlepších fulltextů vůbec. Bohužel v poslední době upadá a je pomalý. Na druhou stranu obsahuje několik zajímavých služeb (hledání obrázků, hudby a multimédií, překlad do jiných jazyků, čistě textová verze, ...).
- Google (<http://www.google.com>) – V poslední době asi nejpoužívanější (obsahuje zajímavé vylepšení prohlížeče – přídatná lišta pro rychlé vyhledávání).
- Excite (<http://www.excite.com>)
- All The Web (<http://www.alltheweb.com>)
- MSN (<http://www.msn.com>)

### 3.2.4 Metavyhledávače

Metavyhledávač je vyhledávací stroj, který využívá ostatní vyhledávače. Po zadání dotazu jej předá vyhledávačům (většinou jich bývá několik), které jej paralelně vyhodnocují, počká si na výsledky a zpracuje je (setřídí, odstraní duplicitní odkazy, zobrazí výsledky v jednotném formátu, ...).

Bohužel ne každý vyhledávač na webu, který o sobě tvrdí že je metavyhledávač, jím skutečně je. Většinou se jedná jen o stránku umožňující zadat dotaz na několik serverů, ale výsledky jednotlivých vyhledávačů se zobrazí na samostatných stránkách, takže základní a nejtěžší funkce (zpracování a sjednocení výsledků z různých zdrojů) není u nich implementována.

Mezi základní výhody těchto systémů patří množství naindexovaných stránek (je to podstatně více než mají normální vyhledávače), uživatelská přívětivost (jednotný interface), rychlost (paralelní zpracování).

U nás existuje zatím jediný metavyhledávač:

- Archon (<http://www.archon.cz>)

Ve světě jich je podstatně více:

- Search.com (<http://www.search.com>)
- Apollo 7 (<http://www.apollo7.de>)
- Meta Crawler (<http://www.metacrawler.com>)
- Monster Crawler (<http://www.monstercrawler.com>)
- Search Caddy (<http://www.searchcaddy.com>)

### 3.2.5 Klientské vyhledávače

Další možností, jak hledat informace na Internetu, jsou takzvané klientské vyhledávače. Jedná se o aplikace nainstalované na lokálním disku, které pro vlastní vyhledávání používají několik (až několik desítek) různých katalogových, fulltextových a jiných vyhledávačů. Jde tedy vlastně o metavyhledávače, které mají podobu klasické aplikace.

Výhod těchto systémů je několik:

- Uživatelské prostředí – jelikož se jedná o aplikaci a ne o webovou stránku, může se spousta rysů velmi snadno nastavit a toto nastavení uložit (což ne každý vyhledávač podporuje).
- Podstatně rychlejší start – aplikace je nainstalována na lokálním disku a na rozdíl od webového vyhledávače není třeba před vlastním vyhledáváním nic stahovat.
- Většina těchto programů uchovává historii vyhledávání včetně výsledků (seznamu nalezených dokumentů). Uživatel se tak může snadno vrátit k předchozím dotazům a tyto dotazy dále „ladit“.
- Další služby – generování reportů, možnost automatického skenování (hledání změn, novinek, dalších dokumentů, ...).

Příklady klientských (meta)vyhledávačů:

- BullsEye  
(<http://www.intelliseek.com/prod/bullseye/bullseye.htm>)
- Copernic (<http://www.copernic.com/>)
- Babylon (<http://www.babylon.com/>)



# Kapitola 4

## Obdobné aplikace

Aplikace, která je součástí diplomové práce, se nejvíce podobá klientským vyhledávačům. V současné době je jich na trhu několik. Jedná se převážně o sharewarové produkty s omezenou funkcností (jsou zakázané pokročilejší operace, nelze použít všechny vyhledávací služby, ale pouze některé, ...).

Níže následuje bližší popis několika aplikací (viz [3]), kterými byla ovlivněna tvorba programu DISClient. Existuje i spousta dalších produktů (Webferret, Search+, ...). Ty však nedosahují takových kvalit a možností, proto zde nebudou uvedeny.

### 4.1 Copernic 2001

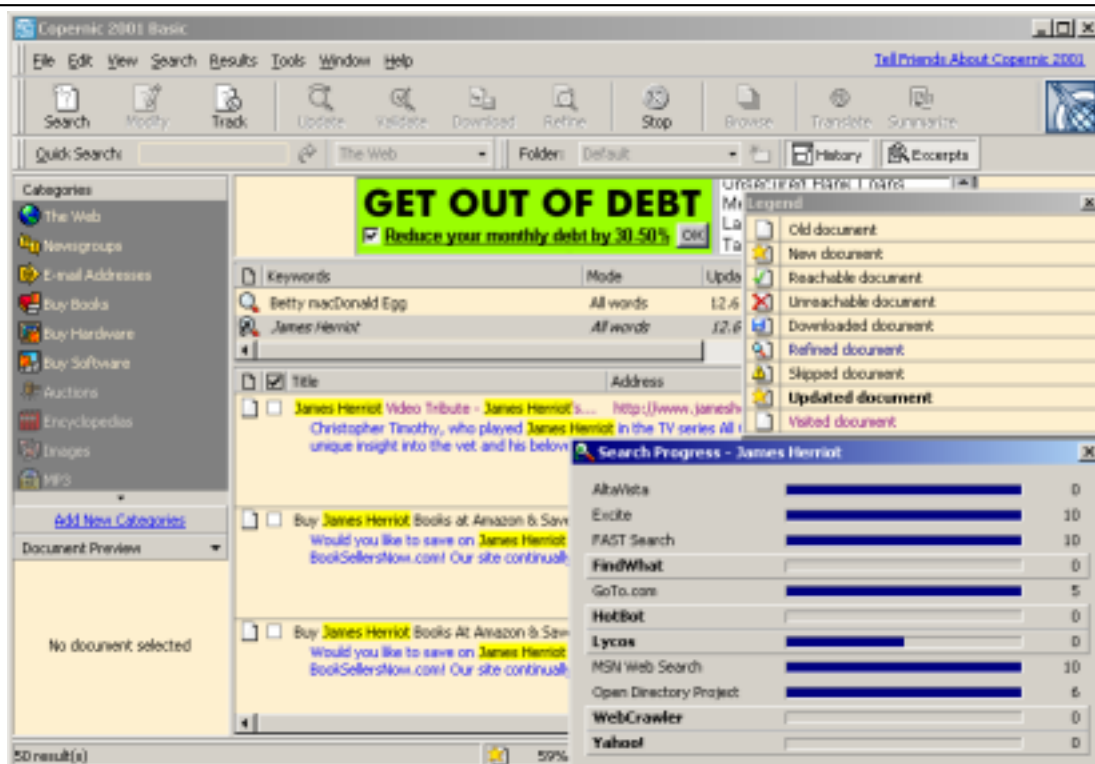
Copernic 2001 je klasický desktopový vyhledávač, jenž pro vlastní vyhledávání využívá asi 80 webových vyhledávacích strojů. Aktuálnost je zajišťována automatickým updatem nastavení vyhledávacích serverů při startu aplikace či vyhledávání. Uživatel se tak nemusí o nic starat (případně může to samé provést příkazem z menu).

Vzhled aplikace (Obrázek 4.1) je víceméně standardní – v levé části seznam kategorií, vpravo seznam dotazů, pod ním přehled nalezených dokumentů k aktuálnímu dotazu a v levém dolním rohu malý náhled vybraného dokumentu.

Při práci je nejdříve třeba vybrat oblast, ve které se bude hledat. K výběru nabízí aplikace více než 40 kategorií. V neregistrované verzi je k dispozici pouze prvních šest: web, diskusní fóra, emailové adresy, nákup knih, hardwaru a softwaru. Po registraci se nabídka rozšíří o mnoho dalších (aukce, zdraví, humor, obrázky, filmy, MP3, recepty, věda, cestování, sport a podobně), jiné se nacházejí na stránkách výrobce.

Po zvolení kategorie je nutno zadat vlastní dotaz. To spočívá v zadání několika termů a zvolení vztahu mezi těmito termy a hledanými dokumenty: zda dokumenty musí obsahovat všechna slova, zda postačí alespoň jedno nebo zda se jedná o přesnou frázi. Zajímavou volbou je mód „Odpověď na otázku“, kdy se místo klíčových slov napíše otázka celou větou a program už si to sám převede (například „Where was Albert Einstein born?“). Tady je nutné podotknout, že celý program je anglicky, používá zahraniční vyhledávače a tedy je nutné tuto otázku položit v angličtině. Také lze upravit seznam vyhledávačů, které se pro vlastní hledání použijí.

Plná verze programu umožňuje automatické odstranění nedosažitelných dokumentů. Vyhledávání pak sice trvá déle (kontroluje se existence nalezených dokumentů), ale uživatel má jistotu, že nalezené dokumenty jsou dostupné.



Obrázek 4.1: Copernic

Posledním krokem editace dotazu je zadání maximálního počtu výsledků od jednotlivých vyhledávacích strojů a počet výsledků celkem.

Po potvrzení se program pustí do vlastního vyhledávání. Průběh se zobrazuje v pomocném dialogovém okně, kde se také vypisuje kolik dokumentů bylo nalezeno jednotlivými kontaktovanými vyhledávači. Výsledky (odpovídající dokumenty) se poté zobrazí v přehledném seznamu.

Nalezené dokumenty lze samozřejmě prohlížet – pro zobrazení lze využít jakýkoliv nainstalovaný prohlížeč a nebo interní prohlížeč této aplikace, který dovede navíc v zobrazovaném dokumentu zvýraznit hledaná slova, snadno přecházet mezi jednotlivými dokumenty a obsahuje i příjemné klávesové zkratky. Ve stejném prohlížeči lze také zobrazit seznam nalezených dokumentů, což se velmi podobá prohlížení výsledků na běžném webovém fulltextu. Vybrané dokumenty lze kromě prohlížení také uložit na disk (včetně obrázků, či bez nich).

Zajímavá je také možnost zobrazit si dokument v jiném jazyce (stačí zadat z jakého do jakého jazyka se má přeložit). K dispozici je angličtina, francouzština, němčina, italština, japonština, portugalská a španělština.

Ve fázi specifikace dotazu sice nelze odfiltrovat nefunkční odkazy (tedy ve freewarové verzi programu), ale po prohledání již toto naštěstí provést lze.

Je-li navrácených dokumentů příliš mnoho, lze je filtrovat. K tomu slouží příkaz Refine, který zkontroluje, zda vybrané dokumenty opravdu obsahují uživatelem zadaná slova (předtím se automaticky stáhnou na pevný disk). Přitom je možné poupravit slova, která má dokument obsahovat (k dispozici jsou spojky AND, OR, EXCEPT, NOT, NEAR) a dotaz takto doladit.

Přehled nalezených dokumentů odpovídající dotazu se dá vyexportovat v několika různých formátech: čistý text, HTML, CSV (Comma Separated Value – atributy jsou odděleny čárkami. Formát podporovaný třeba Excelem), DBF a XML.

Výsledky jednotlivých vyhledávání se ukládají na disk do speciálních adresářů, které lze snadno editovat. Díky tomu se k nim může uživatel kdykoliv později vrátit a dále s nimi pracovat: pozměnit dotaz, znovu vyhledat (přidají se pouze nově nalezené dokumenty), duplikovat a podobně.

Aplikace lze vizuálně upravovat – od zobrazování/schovávání jednotlivých prvků, přes sloupečky seznamů, nastavení výchozích voleb při editaci dotazu až po vlastní skin (uplná změna grafického provedení aplikace).

Při instalaci (případně později v nastavení) tohoto produktu je možno zabudovat několik rozšíření do Internet Exploreru: Přidání odkazu do nástrojové lišty a menu, nové položky do kontextového menu prohlížeče, nahrazení standardního nástroje pro vyhledávání, přidání volby překladu stránek.

Ačkoliv je freewarový produkt odlehčenou (chybí mu několik příjemných funkcí) verzí placeného produktu, je plně funkční. Placená verze „Copernic Plus“ obsahuje navíc přístup k 1000 vyhledávacích strojů v 93 kategoriích a neobsahuje reklamní proužky. V ještě dražší verzi „Copernic Pro“ je k dispozici několik nových funkcí (automatické odstraňování nefunkčních odkazů, vyhledávací agenti, e-mailové upozornění na nové dokumenty odpovídající dotazu a mnohé další).

Název a verze	Copernic 2001 Basic, verze 5.01
Adresa	<a href="http://www.copernic.com">http://www.copernic.com</a>
Výrobce	Copernic Technologies Inc.
Typ	Freeware
Cena	Freewarová verze – \$0, Copernic PLUS – \$39.95, Copernic PRO – \$79.95
Požadavky	486DX 66 MHz (a vyšší), 15 MB RAM, 10 MB volného místa na disku, operační systém Microsoft Windows 95, 98, Me, NT4, 2000, XP, prohlížeč Netscape 3 nebo Internet Explorer 3 (a novější).

Tabulka 4.1: Informace o aplikaci Copernic 2001

Klady:

- Automatická aktualizace přes Internet.
- Možnosti nastavení prostředí.
- Překlad stránek do cizích jazyků.
- Validace dokumentů.
- Vlastní prohlížeč se zvýrazněním hledaných termů.

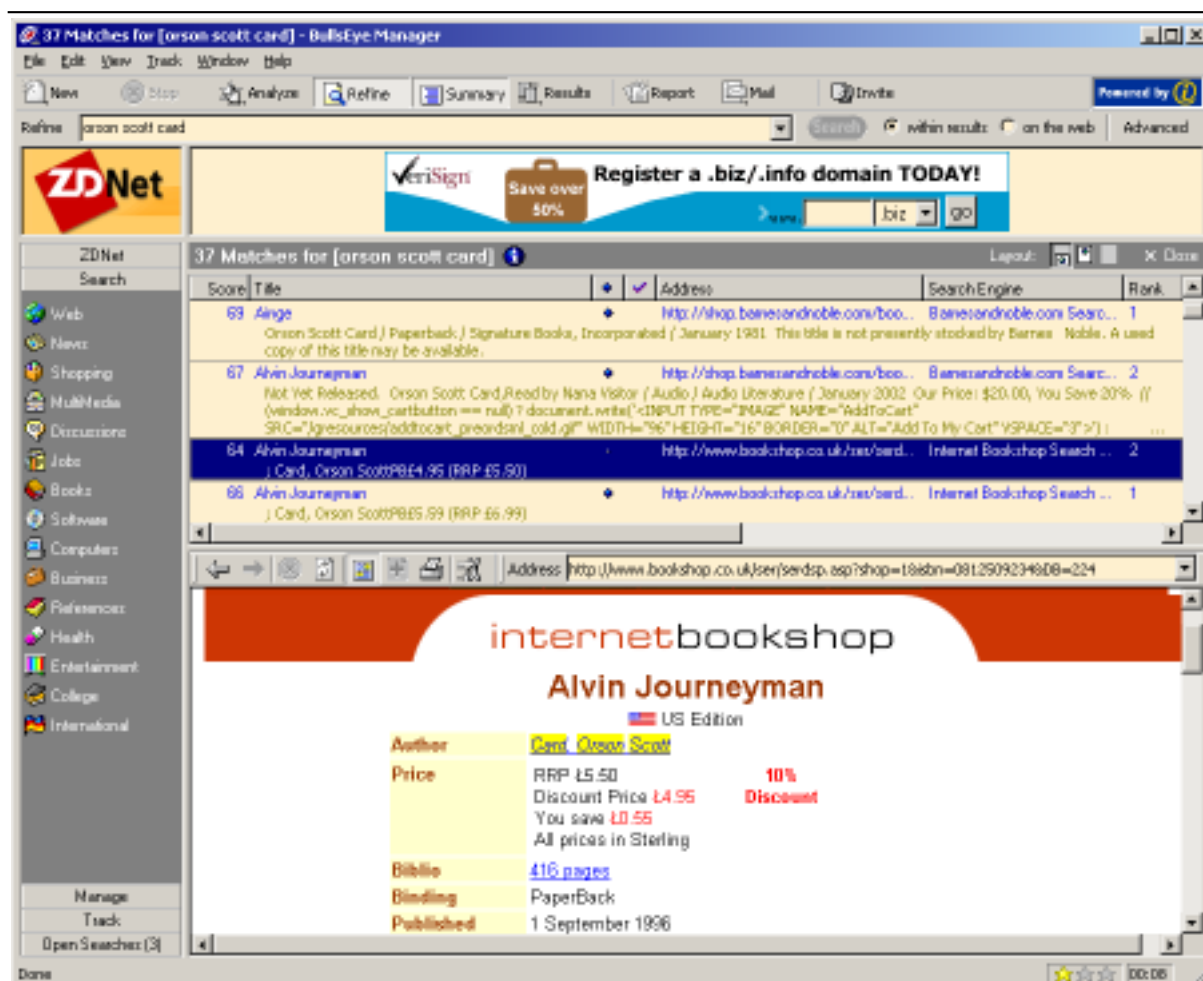
Zápory:

- Dále nerozšiřitelné.
- Omezené možnosti dotazu (nepodporuje spojky a další operátory).

- Podpora a integrace pouze do prohlížeče MSIE.
- Shareware (omezená funkčnost, cena).

## 4.2 Bulls Eye 2

Bulls Eye 2 je přímo „Desktopový vyhledávací portál“. Hlavní okno je rozděleno na několik částí (Obrázek 4.2): Seznam výsledků, náhled aktuálně vybraného dokumentu v tomto seznamu a přehled kategorií, ve kterých je možno vyhledávat – nachází se na levé straně a obsahuje jednotlivé typy vyhledávání: Web, News, Jobs, Books, Software, Health a mnoho dalších.



Obrázek 4.2: Bulls Eye

Po vybrání kategorie při vytváření nového dotazu je třeba zadat hledané termíny (k dispozici je přehled dříve zadávaných termínů). V pokročilém módu lze využít rozšířené nastavení a vytvořit podstatně komplikovanější dotazy za pomoci operátorů AND, OR, NOT, NEAR. Je možné si také prohlédnout a případně pozměnit seznam vyhledávacích webových serverů, které se budou používat a nastavit další parametry (maximální počet výsledků získaných od každého stroje, určit jak mají být výsledky seříděny).

Podle vybrané kategorie lze blíže specifikovat, kde se má hledat (například při vyhledávání knih lze určit, zda se termý týkají názvu, autora, ISBN, zda se hledá knížka v obchodech nebo v knihovnách nebo zda chce uživatel najít volně stažitelnou knihu, ...).

Posledním krokem je volba způsobu analýzy dokumentů – žádný, odstraňování neplatných odkazů nebo kontrola, zda dokument opravdu odpovídá zadanému dotazu (v tom případě je samozřejmě nutné dokument nejdříve stáhnout na lokální disk a prozkoumat, což se provede automaticky).

Po potvrzení dotazu začne probíhat vlastní vyhledávání a v seznamu nalezených dokumentů se začnou zobrazovat první výsledky (časem se dotáhnou a správně zařadí zbývající). V seznamu se uvádí základní informace o nalezených dokumentech (relevantnost, adresa, název stránky, jméno vyhledávače, který ji našel, ...) po přepnutí do detailního módu se zobrazí další informace (datum, velikost a stav dokumentu). Po odkliknutí dokumentu se tento načte a zobrazí pod seznamem s výsledky. Lze nastavit, aby rámec s dokumentem případně se seznamem byl přes celé okno nebo aby byly vidět výsledky a dokument zároveň.

Další možností je otevírat výsledky v asociovaném webovém prohlížeči. Samozřejmě je zvýraznění hledaných slov v prohlíženém dokumentu a tlačítko pro přidání odkazu do záložek.

Za zmínku také stojí informační přehled vyhledávačů, kam se dotaz posílal. Zde je uvedeno které servery nic nevyhledaly a hlavně proč tomu tak je (nic nenašly, nepodařilo se na ně připojit, jsou mimo provoz, ...).

Z vybraných dokumentů si může uživatel vygenerovat report. Při jeho tvorbě je na výběr několik výsledných formátů (html, čistý text, ...). Výsledek se buď uloží na disk, nebo pošle elektronickou poštou.

V levé části okna, kde se vybírá kategorie vyhledávání, se nalézají ještě několik položek. V jedné z nich je seznam právě otevřených projektů (maximálně tři), v další je položka obsahující záložky (a to od obou nainstalovaných prohlížečů – tedy jak Netscape tak MSIE), uložené projekty a již výše zmíněné reporty a historie dotazů.

Další záložka – Track – slouží k automatickému vyhledávání nových dokumentů, které odpovídají zadanému dotazu. Toto je už ale bohužel rozšíření placené verze této aplikace (BullsEye 2 Pro). Odlehčená verze, která je zdarma, zobrazuje reklamní bannery a až na několik náročnějších rozšíření je plně funkční.

Instalace BullsEye 2 přidá několik „zástupců“ do operačního systému pro snazší a rychlejší spouštění aplikace. Konkrétně do nástrojové oblasti hlavní lišty (tool tray), do panelu nástrojů nainstalovaných prohlížečů (MSIE, Netscape) a do Start menu (Start/Search).

Jednou za čas (řádově několik dní) se aplikace zeptá uživatele, zda nemá zkontrolovat výskyt nových aktualizací a případně nabídne jejich stažení a instalaci.

Klady:

- Možnost využívat základní spojky a další parametry při vytváření dotazu.
- Automatický update.
- Podpora e-mailu (zasílání reportu).
- Integrace do systému.

Název a verze	Bulls Eye 2, verze 2.5
Adresa	<a href="http://www.intelliseek.com/prod/bullseye/bullseye.htm">http://www.intelliseek.com/prod/bullseye/bullseye.htm</a>
Výrobce	IntelliSeek Inc.
Typ	Bulls Eye 2 – Freeware, Bulls Eye 3 Pro – Free Trial (15 dní)
Cena	Freewareová verze – \$0, Bulls Eye Plus – \$49.99, Bulls Eye 3 Pro – \$199
Požadavky	Pentium 300 MHz (a vyšší), 64 MB RAM, 100 MB volného místa na disku, operační systém Microsoft Windows 98, ME, 2000, NT4.0 (SP6), prohlížeč Internet Explorer 5.01 SP1 (a novější).

Tabulka 4.2: Informace o aplikaci Bulls Eye 2

- Napojení na oba internetové prohlížeče.
- Přehledné prostředí.

Zápory:

- Pouze 3 současně otevřené dotazy.
- Nerozšiřitelné.
- Nevyužívá plně možností fulltextů.
- Shareware (omezená funkčnost, cena).

## 4.3 Babylon

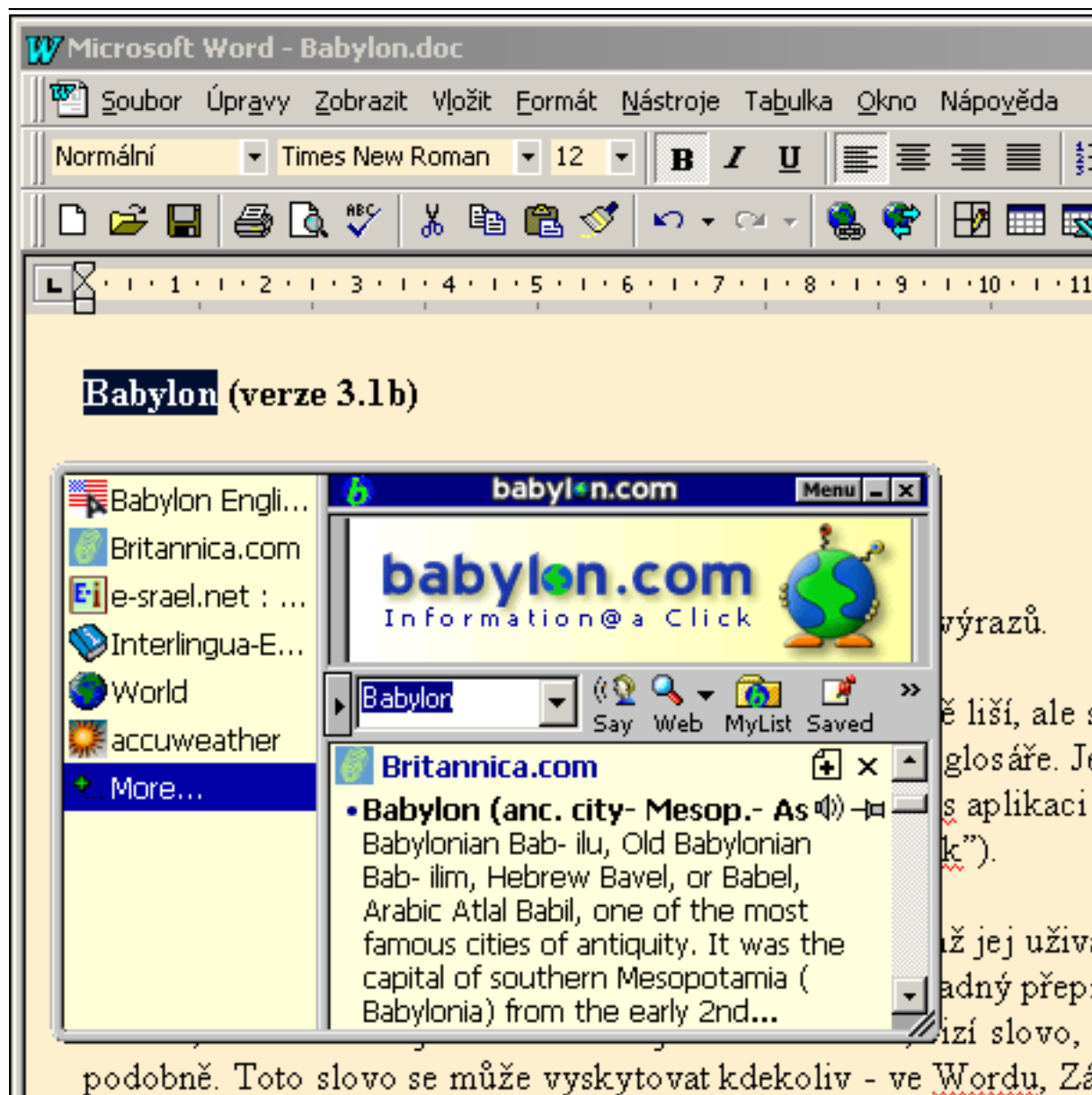
Babylon je se od výše uvedených aplikací poměrně liší, ale stále jde o desktopový vyhledávač, který ale využívá encyklopedie, slovníky a glosáře. Je to interaktivní program, který okamžitě po kliknutí na slově v libovolné windows aplikaci zobrazí příslušnou informaci o tomto výrazu („instant information @ a click“).

Po instalaci se aplikace usídí v nástrojové liště a čeká, až jej uživatel vyvolá klávesomyší zkratkou – pravé nebo prostřední tlačítko myši plus případný přepínač (Ctrl, Alt, Shift) – nad slovem, které chce objasnit. Může se jednat o zkratku, cizí slovo, neznámý výraz, název a podobně. Toto slovo se může vyskytovat kdekoliv – ve Wordu, Zápiscíku, v libovolném dialogu, ba dokonce jako název tlačítka či okna. Po kliknutí se objeví malé okno (Obrázek 4.3), ve kterém se program pokusí objasnit význam daného slova za použití dostupných glosářů.

Kromě glosářů využívá i slovníky překládající mezi hlavními jazyky. Seznam slovníků a glosářů se nachází na webu výrobce, kde jsou přehledně rozříděné do jednotlivých kategorií a podkategorií.

Mezi další funkce, kterými tato aplikace disponuje, patří převody jednotek, zahraničních měn, časových pásem (byl-li například vyvolán nad heslem „24 cm“ nabídne ihned konverzi centimetrů na jiné délkové jednotky) a správná výslovnost uživatelem zkoumaného slova či zkratky.

Primární glosáře a slovníky (ty, které se použijí při prvním vyhledávání významu zkoumaného slova) lze využívat i bez stahování (stačí si je zaregistrovat). Pokud jsou



Obrázek 4.3: Babylon

uložené na disku, pracuje program i bez připojení na Internet – uživatel tím získá rychlejší vyhledávání, ale přijde o nejaktuálnější změny a místo na disku).

Jednotlivé glosáře lze tedy přidávat, ubírat, pokud jsou jen zaregistrované, tak i stáhnout na disk, deaktivovat je a později znovu aktivovat, ohodnotit, případně rovnou napsat autorovi několik poznámek.

Nestačí-li nabízené slovníky, je možnost vytvořit si vlastní a dát jej ostatním k dispozici. K tomu je zapotřebí pouze Babylon Builder, který je zdarma dostupný na stránkách tohoto produktu.

Nebylo-li hledané slovo nalezeno v žádném slovníku (případně není-li uživatel s výsledkem spokojen), může ještě zkusit štěstí v hledání pomocí webových vyhledávačů, které jsou také k dispozici. Výsledkem je zobrazení stránky s výsledky v internetovém prohlížeči.

Nalezené výklady hledaných slov lze pro pozdější použití uložit do speciální složky a snadno se k nim vrátit.

Název a verze	Babylon, verze 3.1b
Adresa	<a href="http://www.babylon.com/">http://www.babylon.com/</a>
Výrobce	Babylon Ltd.
Typ	Free Trial verze, 30 dní
Cena	Babylon-Pro \$17.95

Tabulka 4.3: Informace o aplikaci Babylon

Klady:

- Malá rychlá aplikace, která je kdykoliv k dispozici, snadná aktivace.
- Možnost vytvářet vlastní slovníky.

Zápory:

- Zaměřené pouze na výklad pojmu, slovníky.
- Shareware (omezená doba funkčnosti, cena).



# Kapitola 5

## Programátorská dokumentace (implementace)

### 5.1 Vývoj architektury

Původní návrh aplikace byl poměrně jednoduchý a počítal s tím, že se vlastní program bude skládat z uživatelského prostředí a DLL knihoven, které budou sloužit pro komunikaci s jednotlivými servery (Obrázek 5.1).

Tyto knihovny budou napojeny na cílové DIS servery, případně webové vyhledávací služby typu Altavista, Google, Serge a další. Propojení mezi vlastním uživatelským prostředím a DLL knihovnami zajišťuje společný interface obsahující základní objekty pro komunikaci.

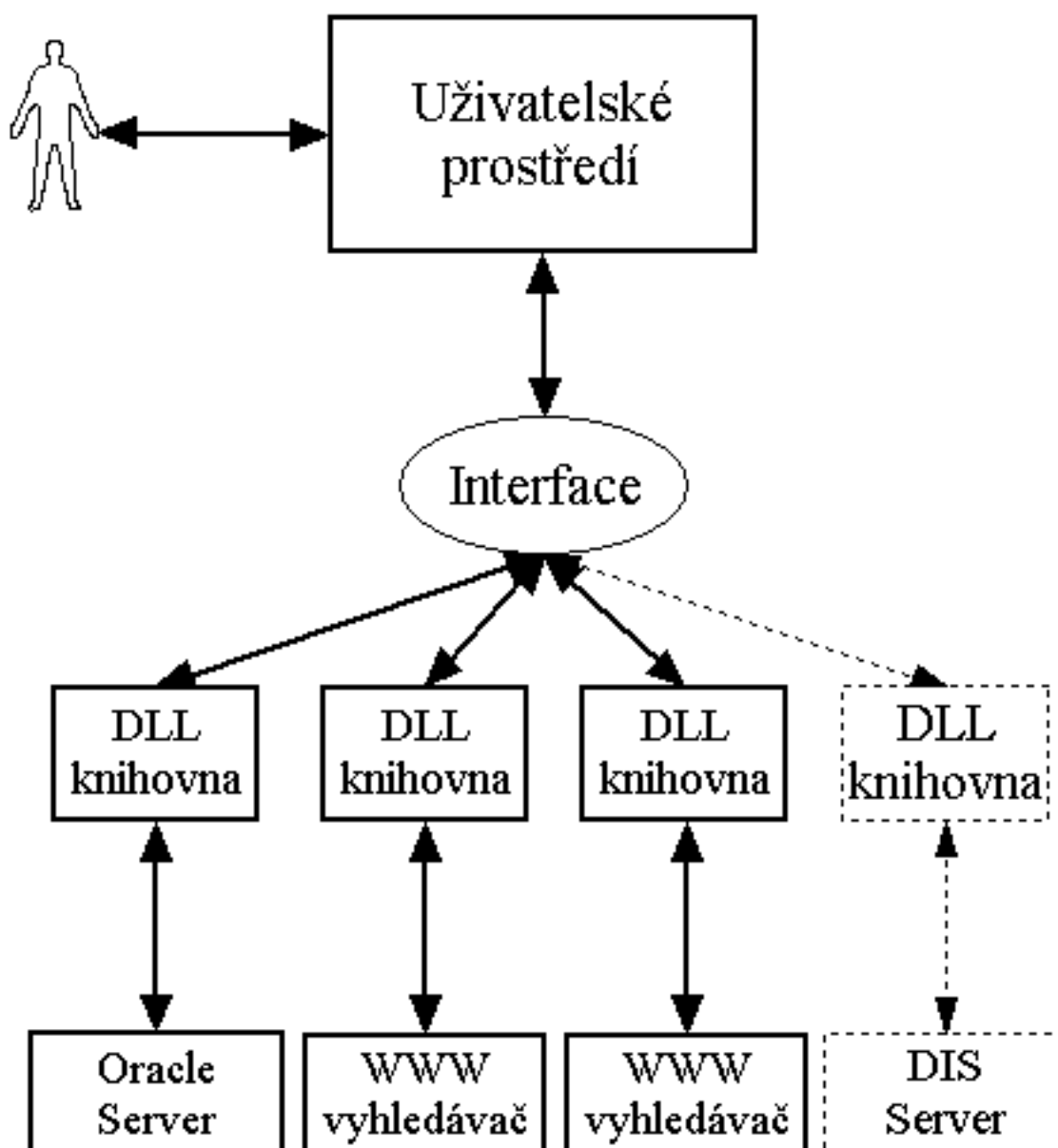
Aby byl systém dále snadno rozšiřitelný, budou DLL knihovny linkovány dynamicky, tedy pro rozšíření možností aplikace o další cílový server stačí dodat příslušnou DLL knihovnu implementující funkce rozhraní (zejména jde o definici parametrů nastavení serveru, dotazu, vlastní vyhledávání, získání dokumentu, definování způsobu zobrazení dokumentu atd.).

Později se tato architektura ukázala nevýhodná zejména z důvodu dalšího využití. Proto byla vlastní aplikace rozdělena do dvou modulů (Obrázek 5.2).

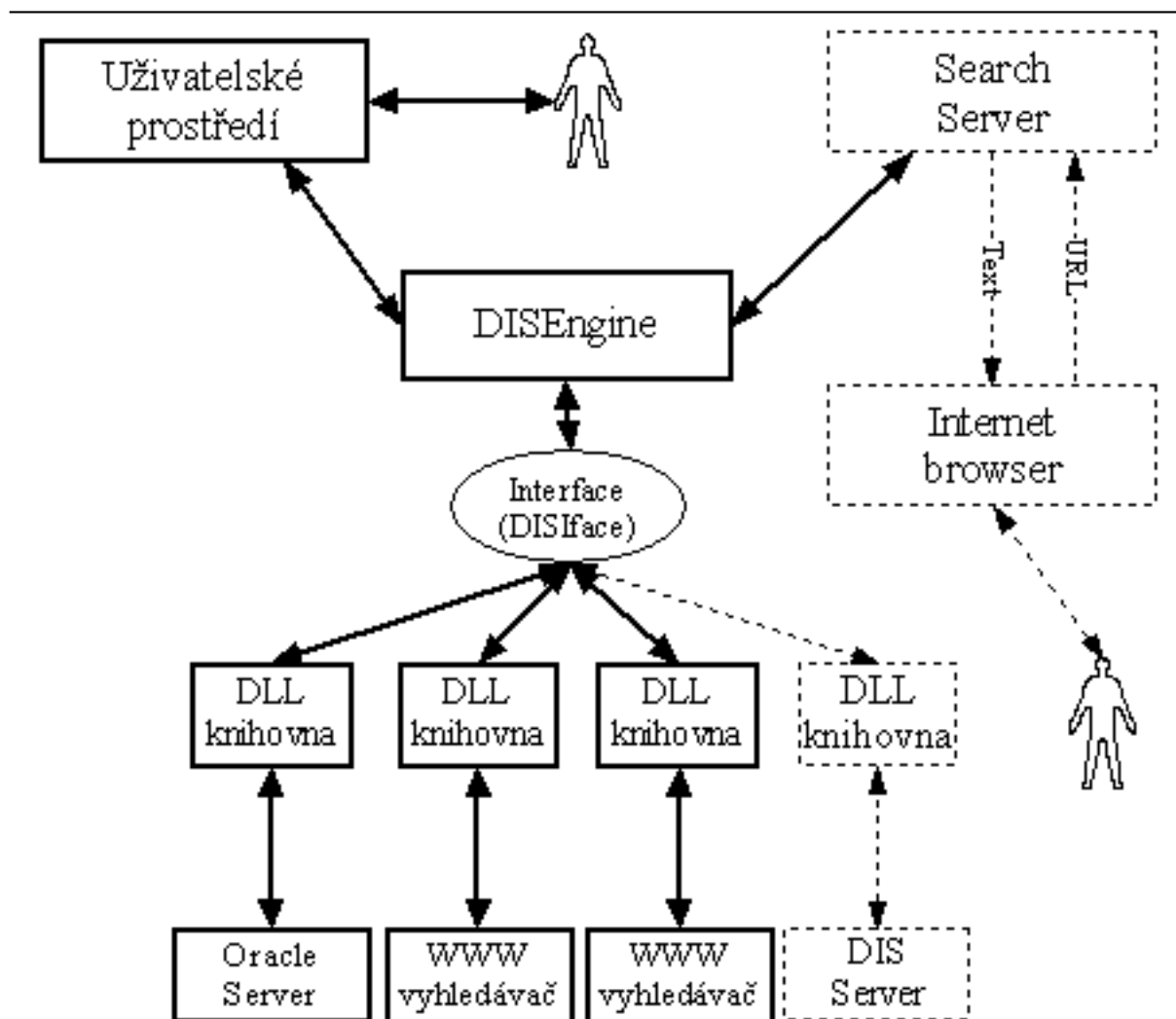
**DISEngine** Vlastní jádro, které se stará o komunikaci s DLL knihovnami, vytvoření dotazu využívající zadané servery, vyhledávání příslušných dokumentů na základě parametrů dotazu, ukládání a načítání dotazů a dokumentů na disk, . . .

**GUI** Grafické uživatelské rozhraní zajišťuje komunikaci s uživatelem. Jedná se o přívětivé uživatelské prostředí, které zpřístupňuje uživateli aplikace veškeré možnosti jádra (DISEngine). V budoucnu místo tohoto modulu může být implementován jiný.

Díky tomuto rozdělení lze snadno vytvořit jiný modul chovající se jako webová služba. Na rozdíl od běžných metavyhledávačů nebude využívat pouze klasické webové vyhledávače, ale i další servery (Oracle a jiné databáze, soubory na disku, e-mailové konference uložené v poštovním klientu, . . .). Tento modul bude přijímat požadavky na vyhledávání, které přijdou ze sítě. Vlastní dotaz bude obsažen v části URL adresy a jako odpověď se bude zasílat HTML stránka obsahující seznam nalezených dokumentů.



Obrázek 5.1: Výchozí architektura aplikace

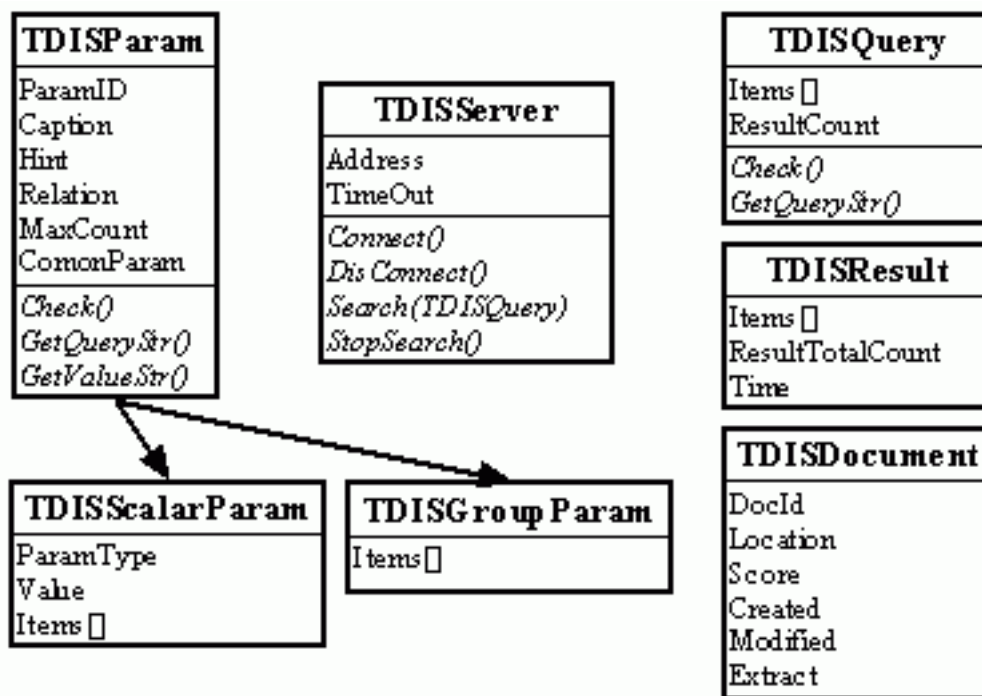


Obrázek 5.2: Finální architektura aplikace

## 5.2 Popis rozhraní

Rozhraní (interface) mezi jádrem aplikace (DISEngine) a uživatelským prostředím tvoří knihovna definující objekty, které si tyto moduly předávají.

Prapůvodní interface byl značně jednoduchý a nedomyšlený (Obrázek 5.3).



Obrázek 5.3: Prapůvodní popis rozhraní

Finální verze rozhraní obsahuje mnoho nových objektů, které se v první verzi nevykytovaly (Obrázek 5.4 a 5.5).

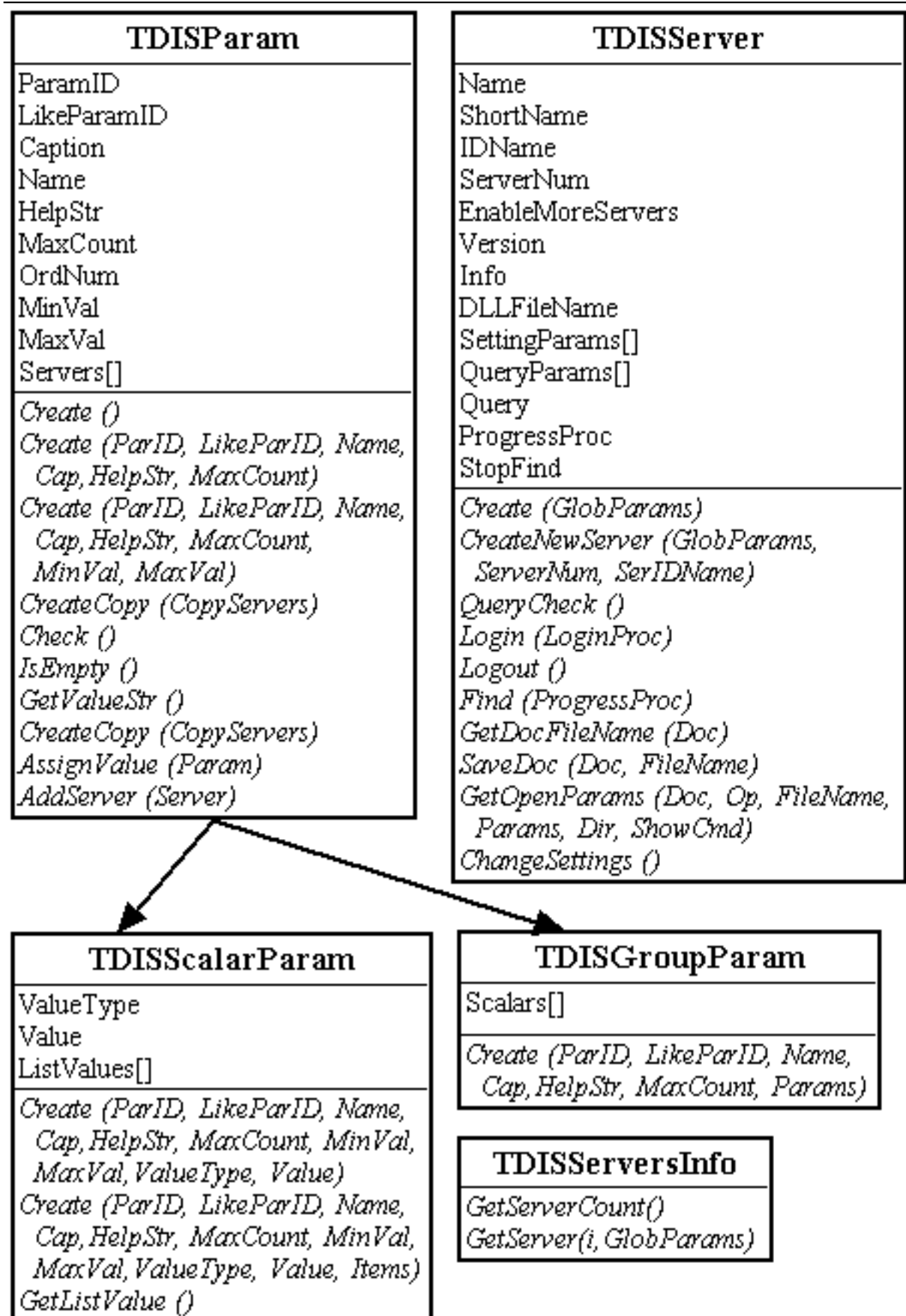
Bližší vysvětlení funkcí jednotlivých komponent rozhraní následuje v dalším textu.

### 5.2.1 Systém parametrů

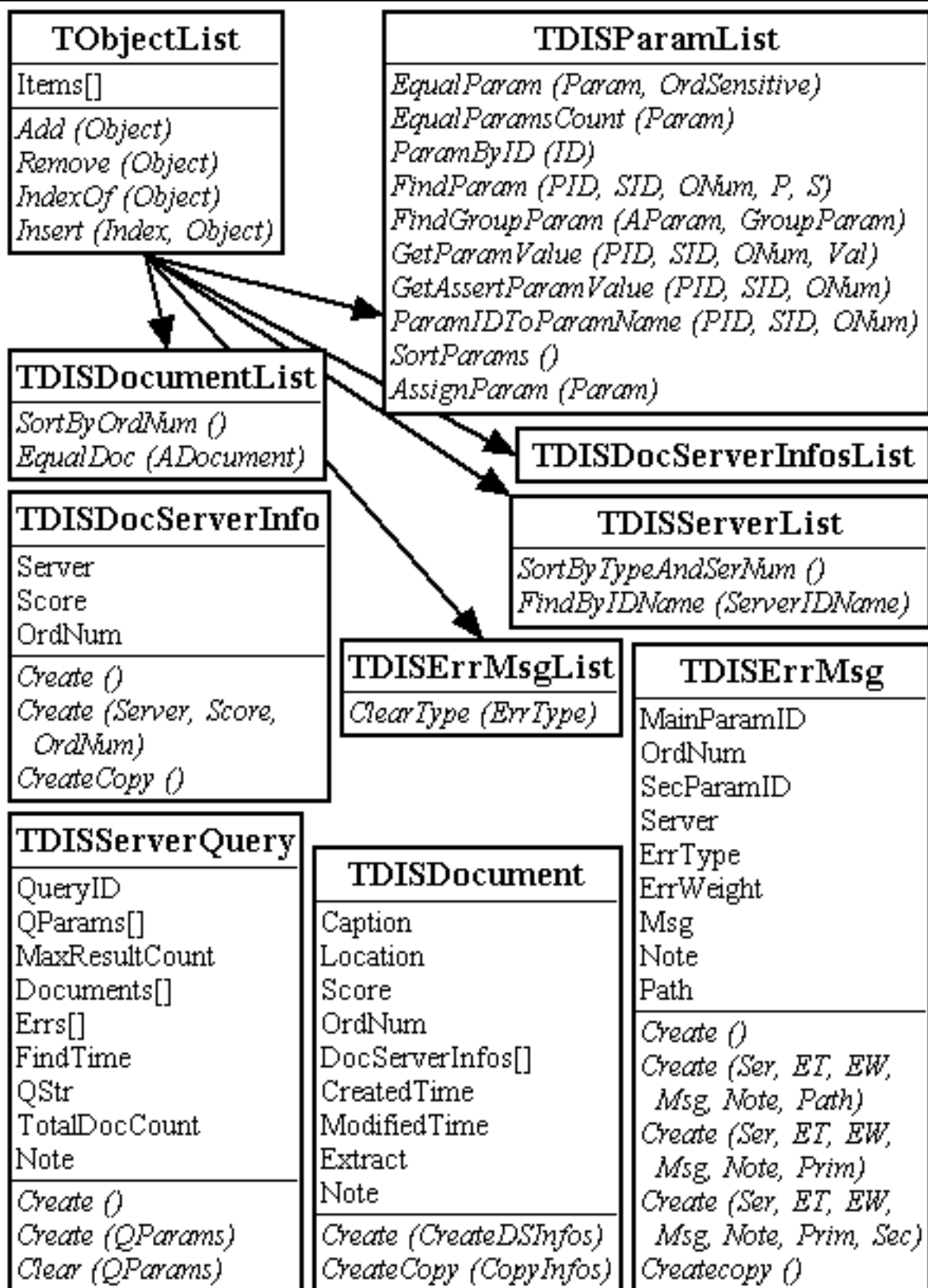
Pro specifikaci dotazů a nastavení jednotlivých serverů bylo třeba vybudovat systém parametrů. Bylo několik alternativ, jak tento systém vytvořit:

- Obecný systém – v jádře aplikace by byly nadefinované všechny možné parametry dotazu, které mají smysl. Při editaci by se uživateli tyto parametry zobrazily a on by je vyplnil. Seznam by pak dostaly všechny servery a ty by využily jen parametry, které samy podporují.

Nevýhody tohoto systému jsou zřejmé – málo flexibilní, moc omezené a dále nerozšiřitelné. Na jednu stranu by systém obsahoval zbytečné množství parametrů, které žádný server nevyužije (a uživatel by je tedy vyplňoval zcela zbytečně). Na stranu druhou by tento systém nebyl úplný – vždy by se našel server s parametrem, který tento systém neobsahuje.



Obrázek 5.4: Objekty rozhraní



Obrázek 5.5: Objekty rozhraní

- Parametry definované v DLL knihovnách. Tento systém by byl velmi flexibilní, bylo by možné nadefinovat libovolný parametr. Nevýhodou je, že každá DLL knihovna bude zbytečně znovu definovat existující parametry.

V tomto případě jsou dvě možnosti, jak definovat seznam parametrů pro uživatele. První možností je udělat průnik parametrů, které nabízí uživatelem vybrané servery, druhou možností je vytvořit jejich sjednocení.

Nevýhodou je nejednoznačnost – dva servery můžou definovat jeden parametr, který defacto plní stejnou funkci, různě. Uživatel pak bude nucen tento parametr vyplňovat dvakrát.

- Konečné řešení – je založeno na využití kladných vlastností výše uvedených možností. Parametry budou definovány v DLL knihovnách, ale určitá množina nejčastějších a nejobecnějších parametrů bude definována mimo a bude volně využitelná všemi knihovnami.

Parametry budou jednoznačně identifikovány a tak nebude problém vytvořit rozumné sjednocení parametrů různých serverů bez ohledu na místo vytvoření parametru, bez toho aby se v tomto sjednocení objevily duplicity.

Základem je abstraktní objekt `TDISParam`, od něhož je odvozen objekt reprezentující parametr obsahující vlastní hodnotu (`TDISScalarParam`). Druhý odvozený objekt (`TDISGroupParam`) slouží k seskupení několika parametrů popisujících stejnou vlastnost do skupiny.

### 5.2.1.1 TDISParam

Mezi základní atributy objektu patří `ParamID`, který jednoznačně identifikuje parametr. `LikeParamID` se využívá při zatřídění parametru do seznamu při jeho vizuálním zobrazení. `Caption`, `Name` a `HelpStr` slouží k popsání významu vlastního parametru (nebo skupiny).

Některé parametry lze duplikovat. Jsou to ty, které mají `MaxCount` větší než nula. `OrdNum` uvádí číslo kopie parametru, originální (původní parametr, který byl duplikován) parametr má `OrdNum = 1`.

Pole serverů `Servers` obsahuje seznam serverů, které tento parametr podporují (pouze u parametrů na straně aplikace, na straně serveru je tento atribut nastaven na hodnotu `nil`).

### 5.2.1.2 TDISScalarParam

Tento potomek `TParam` obsahuje vlastní hodnotu. Ta je obsažena v řetězcovém atributu `Value`. Parametr může obsahovat hodnotu různého typu (`ValueType`):

`vtString`, `vtPassString` Řetězec, případně zašifrovaný řetězec (pro uložení hesla a jiných údajů, které se nesmí zobrazovat).

`vtBoolean` Pravdivostní hodnota, je uložena jako řetězec „1“ případně „0“.

`vtNumber`, `vtDecNumber` Číslo, případně číslo s desetinnou čárkou.

`vtDate` Datum, který je v řetězcovém atributu `Value` uložen jako řetězec reprezentující číslo s desetinnou čárkou (stejně jako typ `TDateTime` obsahuje toto číslo před desetinnou čárkou počet dní od roku 1899 a za čárkou část dne).

`vtList` Seznam, tedy výběr z předdefinovaných hodnot, které se nachází v poli `ListValues`. Vlastní hodnota obsahuje index vybraného prvku.

### 5.2.1.3 TDISGroupParam

Obsahuje pole `Scalars` dále nedělitelných parametrů typu `TDISScalarParam`. Tyto parametry spolu souvisí, proto jsou uvedeny ve skupině (například jméno a heslo pro přihlášení na vzdálený server, nebo datumové rozpětí pro uložení poslední změny dokumentu).

### 5.2.1.4 Implementace parametrů

Pro snazší vytváření parametrů obsahuje interface několik konstruktorů. Parametry mohou být vytvářeny kdekoliv. Některé parametry, které jsou nejběžnější (a tedy pro většinu serverů společné) jsou definovány ve zdrojovém souboru `GlobalParams`. Pole těchto globálních parametrů předává jádro aplikace jednotlivým serverům při jejich vytváření. Tyto servery si tak snadno mohou některé z nich přidat (vytvořit si svou kopii) do svého seznamu. Tím je zajištěna jistá jednoznačnost základních parametrů.

Další možností, kde vytvářet parametry, je vlastní server. Tady vznikl problém s vývojovým prostředím (Delphi 6 firmy Borland), neboť základní metoda objektu `TObject`:

```
function InheritsFrom(AClass: TClass): Boolean;
```

nepracovala správně.

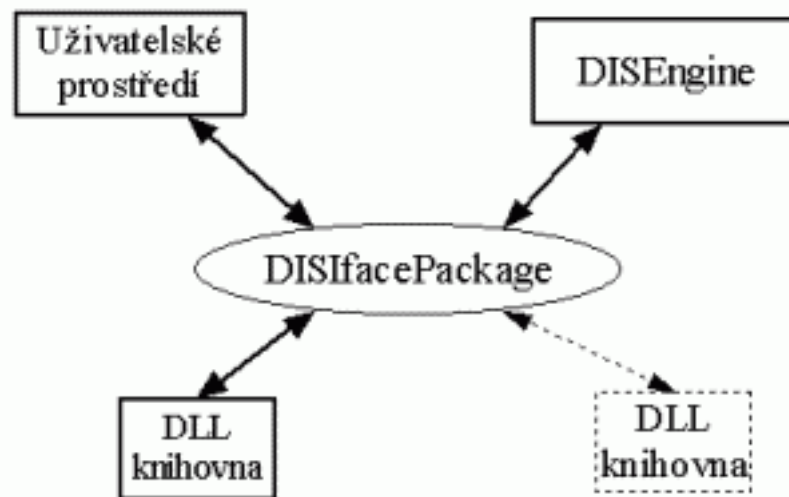
V případě, že byla tato metoda volána v jiném procesu než byl objekt vytvořen (například pokud byl objekt vytvořen v DLL knihovně a pak se testoval jeho typ v aplikaci), vracela chybné výsledky. Řešením bylo zavedení balíčku obsahující vlastní rozhraní (`DISIFacePackage.bpl`), tedy definice všech objektů. Tento balíček musí být využíván jak aplikací, tak všemi DLL knihovnami (Obrázek 5.6).

Cílem implementace bylo, aby systém parametrů umožňoval pro libovolný server zadat libovolný parametr, tedy aby bylo možné plně využít serveru. Na druhé straně byl měl ale zajistit uživatelskou přívětivost. Rozhodně by neměl nutit uživatele vyplňovat pro každý server parametry, které jsou shodné (například termy, které musí/nesmí hledaný dokument obsahovat, jazyk dokumentu, ...). Proto byla implementována metoda na slévání poli parametrů (`GetCommonParams`).

Výsledkem slévání je pole parametrů, které obsahuje všechny parametry všech zvolených serverů, ale pokud servery obsahují stejné parametry, jsou tyto ve výsledném seznamu uvedeny pouze jednou. Každý parametr v poli obsahuje seznam serverů (`Servers`), které daný parametr dodaly. Toto pole ukazatelů poté slouží ke zpětnému nastavení hodnot na stranu serverů – DLL knihoven (`SetCommonParams`).

Nejdůležitější je funkce, která rozhoduje o tom, zda dva parametry jsou si rovny (a tedy budou slity do jednoho):





Obrázek 5.6: Využití balíčku DISIfacePackage

```

function EqualParams (P1, P2: TDISParam;
  OrdSensitive: Boolean): boolean;

```

P1 a P2 jsou porovnávány parametry, `OrdSensitive` určuje, zda se má při porovnávání brát v úvahu jejich atribut `OrdNum`.

Dva parametry jsou shodné, pokud:

- Jsou stejného typu (`TDISScalarParam`, `TDISScalarGroup`).
- Mají stejné `ParamID`.
- Jedná-li se o `TDISScalarParam`, musí mít shodný typ hodnoty (`ValueType`).
- Jedná-li se o `TDISScalarGroup`, musí mít shodný počet skalárních parametrů a tyto musí být shodné.

Tato definice mimo jiné umožňuje, aby dvě různé DLL knihovny implementovaly shodné parametry bez toho, aby sdílely společný kód. Stačí znát `ParamID` a typ parametrů, což lze snadno zjistit přímo v aplikaci.

### 5.2.1.5 Pole parametrů

Pro práci s parametry bylo třeba vytvořit objekt, který by obsahoval více parametrů. K tomu je určen `TDISParamList` – potomek objektu Delphi `TObjectList`. Podobně byly odvozeny objekty pro reprezentaci polí (seznamů) dalších používaných objektů (serverů, dotazů, dokumentů, chyb, ...): `TDISServerList`, `TDISQueryList`, `TDISDocumentList`, `TDISErrorList`, `TDISDocServerInfosList` a také objekt `TDISQueryServerInfosList`. Byly implementovány další potřebné metody pro vyhledávání, třídění a podobně.

Objekt `TDISParamList` obsahuje několik metod, které stojí za povšimnutí.

```
FindParam(PrimID, SecID, OrdNum: integer;  
          var Prim, Sec: TDISParam): boolean;
```

Slouží k hledání parametru daných vlastností (`PrimID`, `SecID`, `OrdNum`). `PrimID` obsahuje ID hledané skupiny, nebo skalárního parametru, který není ve skupině. `SecID` se použije pro hledání parametru ve skupině. V takovém případě `PrimID` obsahuje ID skupiny a `SecID` identifikátor skalárního parametru. `OrdNum` udává pořadové číslo. V `Prim` bude navrácen ukazatel na hledaný skalární parametr (pokud není ve skupině), případně na skupinu. `Sec` se použije v případě hledání skalárního parametru, který se nachází ve skupině (ukazatel na skupinu bude v `Prim`).

```
GetParamValue (PrimID, SecID, OrdNum: integer;  
              var Val: string): boolean;
```

Pokusí se najít zadaný parametr a v proměnné `Val` vrátí jeho hodnotu.

```
function EqualParam(AParam: TDISParam; OrdSensitive:  
                   Boolean = true): TDISParam;
```

Nalezne první parametr shodný s `AParam` dle výše uvedené definice v seznamu.

### 5.2.1.6 Příklady parametrů

Parametr pro zadání slov, která má hledaný dokument obsahovat (nejběžnější parametr, který používá každý server).

```
TDISScalarParam.Create (  
  gpsidMainQueryAnd,  
  gpsidMainQueryAnd,  
  'Musí obsahovat',  
  'Musí obsahovat',  
  'Slova, která dokument musí obsahovat.',  
  1,  
  NotCheckVal,  
  NotCheckVal,  
  vtString,  
  ''  
);
```

Parametr pro zadání časového intervalu, kdy byl dokument vytvořen, případně modifikován. Jelikož musí parametr obsahovat dva časové údaje, nejedná se o jeden skalární parametr, ale o skupinu parametrů. Skupina obsahuje dva skaláry, které je nejprve třeba vytvořit.

```

Scal1 := TDISScalarParam.Create (
    gpsidDocDate_From,
    gpsidDocDate_From,
    'Datum od',
    'Od',
    'Počátek intervalu, kdy byl dokument vytvořen',
    1,
    NotCheckVal,
    NotCheckVal,
    vtDate,
    ''
);
Scal2 := TDISScalarParam.Create (
    gpsidDocDate_To,
    gpsidDocDate_To,
    'Datum do',
    'Do',
    'Konec intervalu, kdy byl dokument vytvořen',
    1,
    NotCheckVal,
    NotCheckVal,
    vtDate,
    ''
);

TDISScalarGroup.Create (
    gpgidDocDate,
    gpgidDocDate,
    'Datum dokumentu',
    'Datum modifikace dokumentu',
    'Časový interval, kdy byl dokument vytvořen',
    1,
    [Scal1, Scal2]
);

```

## 5.2.2 Server

### 5.2.2.1 Struktura DLL knihovny

DLL knihovna implementující DIS server musí exportovat funkci bez parametrů, která vrací potomka objektu `TDISServersInfo`. Tento objekt (`TDISServersInfo`) slouží ke zjištění počtu serverů, které tato knihovna implementuje (metoda `GetServerCount`),

a také k jejich vytvoření metodou `GetServer (i, GlobParams)`, kde `i` je index vytvářeného serveru a `GlobParams` je pole základních nejčastěji používaných parametrů, které server může využít (implementovat).

Z výše uvedeného vyplývá, že jedna DLL knihovna může obsahovat několik serverů, což je velmi výhodné. Obzvláště v okamžiku, kdy jsou tyto servery velmi podobné a mohou se tak snadno odvodit od společného předka (jak to bylo provedeno v případě implementace webových fulltextových vyhledávačů).

### 5.2.2.2 Vlastní server

Vlastní server je v DLL knihovně implementován potomkem objektu `TDISServer`. Objekt obsahuje jednoduché atributy popisující server: jméno (`Name`), jeho zkratku používanou ve stručnějších výpisech (`ShortName`), číslo verze (`Version`) a další informace (`Info`).

Pro vlastní nastavení serveru slouží pole parametrů `SettingParams`. Obdobné pole `QueryParams` slouží k uchování parametrů dotazu, jež server poskytuje. Při vlastním vyhodnocení dotazu se používá instance objektu `TDISServerQuery`. Průběh vlastního vyhledávání se zobrazuje pomocí feedbackové funkce `ProgressProc`. Pokud si uživatel přeje zastavit vyhledávání, nastaví se atribut `StopFind` na `True`.

Někdy je potřeba používat několik serverů stejného typu. Například v případě dat uložených na několika serverech Oracle či InterMedia. Toto je řešeno vytvořením více kopií serverů, které se budou lišit v nastavení (budou mít různé adresy, jména, hesla, ...) a také v některých parametrech dotazu. O tom, zda server podporuje vytváření dalších kopií rozhoduje jeho vlastnost `EnableMoreServers`. Tyto kopie se pak liší ve svém pořadovém čísle (`ServerNum`). Pro jednoznačnou identifikaci serveru slouží atribut `IDName`.

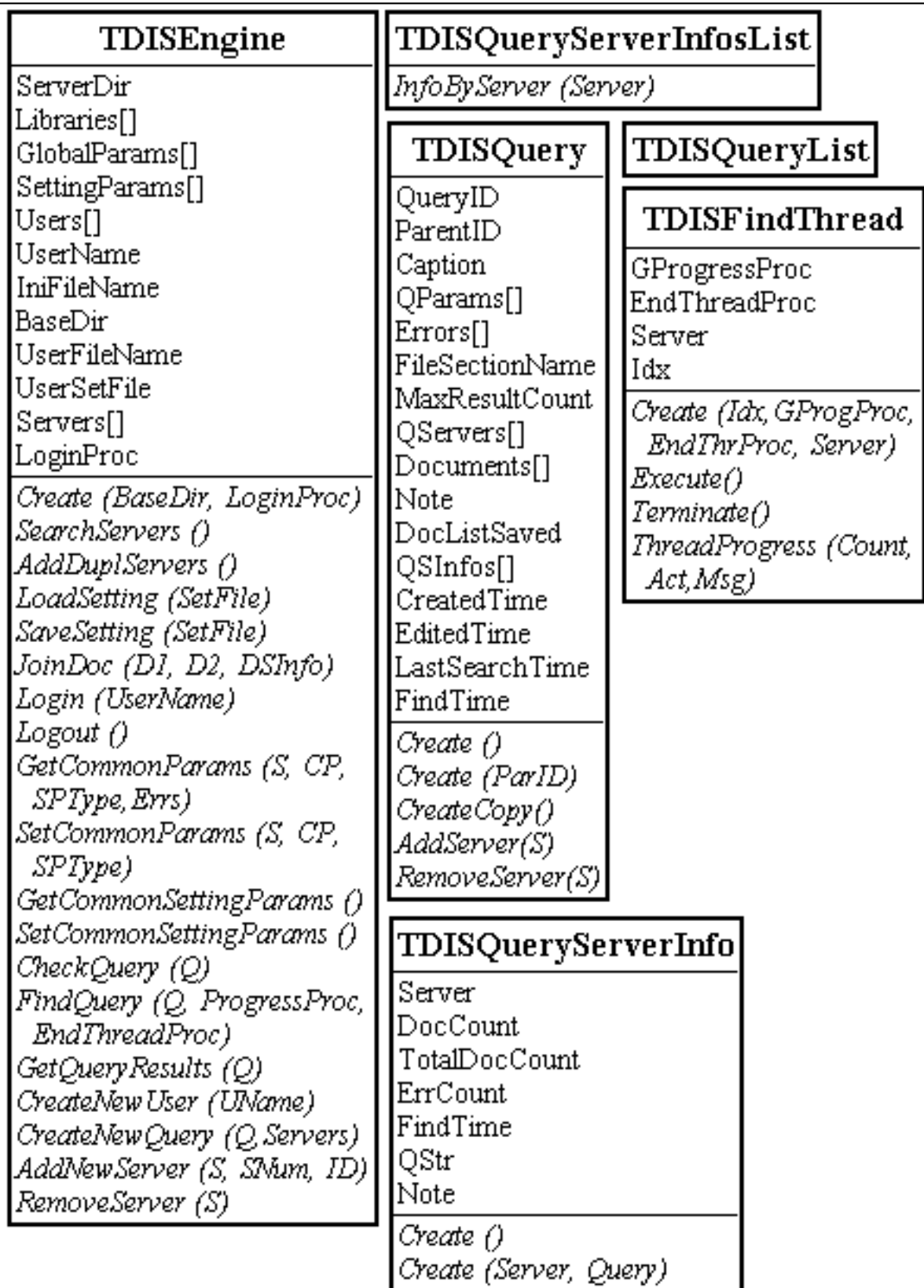
Z metod stojí za zmínku `Login` a `Logout` pro připojení a odpojení od skutečného DIS serveru, `QueryCheck` pro otestování položeného dotazu, a `Find` pro vlastní hledání.

Po získání seznamu dokumentů odpovídajících dotazu by měl sever umět stáhnout dokumenty na lokální disk a zobrazit je. Slouží k tomu metody `GetDocFileName` (zjištění jména dokumentu z objektu `TDISDocument`), `SaveDoc` (získání dokumentu a jeho uložení na disk) a `GetOpenParams` (nastavení parametrů pro jeho zobrazení v asociované aplikaci funkcí `ShellExecute`).

## 5.3 Průběh položení dotazu

Jádro aplikace (`DISEngine`, jeho objekty viz Obrázek 5.7) při přihlášení uživatele vytvoří seznam dostupných DLL knihoven (tyto se musí nacházet v podadresáři `Servers`). Jedna knihovna může obsahovat i několik různých serverů. Jejich počet a instance se získají přes potomka objektu `TDISServersInfo` (viz Oddíl 5.2.2.1). Některé servery mohou mít navíc další kopie (viz Oddíl 5.2.2.2).

Z těchto serverů se získá jejich pole nastavovacích parametrů (`SettingParams`), ze kterých se poskládá globální seznam parametrů nastavení všech serverů. Tento seznam neobsahuje žádné hodnoty. Ze sekce `Global/ServersSetting` ini souboru uživatele (`User.ini`) se načtou všechny dříve uložené hodnoty tohoto seznamu. Nastavené

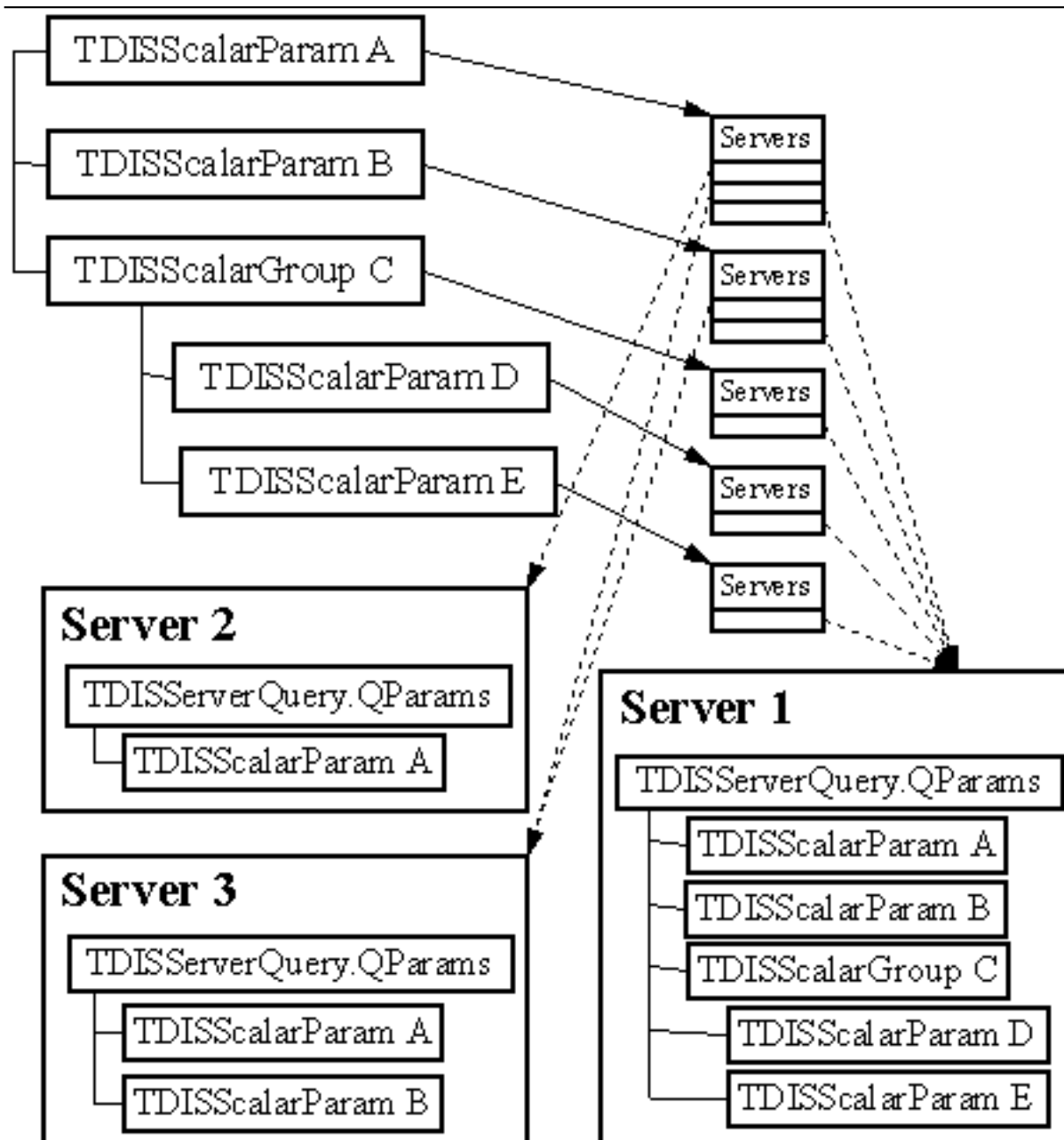


Obrázek 5.7: Objekty jádra aplikace (DISEngine)

parametry se překopírují do lokálních polí příslušných serverů. Nyní je aplikace připravena pro vyhodnocování uživatelských dotazů.

Pro vytvoření prázdného dotazu je nutno zavolat metodu `CreateNewQuery` objektu `TDISEngine`. Prvním parametrem je identifikátor dotazu, ze kterého se odvozuje (odvození slouží k překopírování hodnot z původního dotazu, dojde vlastně k vytvoření kopie dotazu), případně `nil`, pokud se jedná o zcela nový dotaz.

Druhým parametrem je seznam serverů, které má dotaz využít. Tyto servery se překopírují do příslušného seznamu `QServers` v nově vytvořeném objektu `TDISQuery`. Posledním krokem vytvoření nového dotazu je získání seznamu parametrů. Ten se získá zavoláním metody `GetCommonParams`.



Obrázek 5.8: Sestavení pole parametrů dotazu

Nyní je na uživateli, aby vyplnil vlastnosti a další parametry dotazu. Při tom má stále možnost přidávat (`TDISQuery.AddServer`) a ubírat (`TDISQuery.RemoveServer`) servery, kterých se bude dotaz týkat. Po každé změně seznamu serverů je třeba přidat nové parametry, či naopak odstranit přebytečné. Tady je ještě nutno poznamenat, že při přidávání serveru je nejdříve nutné se na tento server připojit voláním jeho metody `Login`. U některých serverů, kde není potřeba se přihlásit (například webové vyhledávače), je tato metoda prázdná. Pokud je volání neúspěšné, server nebude přidán.

Po ukončení editace se zkontrolují uživatelem zadané údaje za pomoci metody jádra `TDISEngine.QueryCheck`. Jelikož je kontrola parametrů závislá na zvolených serverech, vyplní tato metoda serverové objekty `Query` parametry dotazu a poté zavolá `QueryCheck` jednotlivých serverů a získá pole případných chyb (`TDISErrMsgList`), které zobrazí uživateli.

Vlastní vyhledávání je poměrně jednoduché. Nastaví se parametry dotazu na straně serveru (objekt `Query` typu `TDISServerQuery`) a pro každý server se vytvoří hledací vlákno (`TDISFindThread`) s parametry: pořadí serveru, zpětně volaná funkce pro zobrazení postupu hledání `GProgressProc`, další zpětně volaná funkce, která se volá při skončení vyhledávání `EndThreadProc` a odkaz na příslušný server. Hledací vlákno okamžitě po svém vytvoření začne vyhledávat, což spočívá v zavolání metody `Find` přiděleného serveru a předání funkce na zobrazování průběhu. Při ukončení aktivity vlákna (vyhledávání) se zavolá metoda `EndThreadProc`. Průběh hledání lze také předčasně ukončit nastavením atributu `StopFind = True` serveru.

Po ukončení hledání server připraví pole nalezených dokumentů (ne vždy je to tak triviální – viz [Oddíl 5.5.1](#)). Pokud cílový DIS server podporuje ohodnocení, musí být toto ohodnocení převedeno na procenta.

Jakmile všechny servery (vlákna) ukončily vyhledávání, přichází na řadu další fáze, kterou je získání výsledků. Metoda `GetQueryResults` projde všechny oslovené servery a vytvoří si informační objekt typu `TDISQueryServerInfo`, který popisuje výsledek vyhledávání daného serveru (počet nalezených dokumentů, počet vrácených dokumentů, počet chyb, dotazovací řetězec, dobu trvání vyhodnocení dotazu a další).

Vrátí-li server nějaké dokumenty, přidá je do seznamu všech nalezených dokumentů. Ale nejdříve zjistí, zda už daný dokument není v seznamu obsažen (pomocí metody `TDISDocumentList.EqualDoc`). Je-li již v seznamu, spojí dokumenty do jednoho. Tedy přepočítá výsledné skóre a pořadí dokumentu a do pole `DocServerInfos` dokumentu přidá odkaz na další server.

Není-li v seznamu ještě obsažen, přidá jej tam a do pole `DocServerInfos` dokumentu vloží odkaz na informující objekt `TDISDocServerInfo` (kde je uveden odkaz na server, skóre a pořadí dosažené na tomto serveru).

Zde nastal problém se způsobem uspořádávání dokumentů na výstupu. Jelikož ne každý server vrací ohodnocení dokumentu (míru jeho relevantnosti vzhledem k položenému dotazu), nelze použít skóre (jak by se potom měly zařadit neohodnocené dokumenty?). Jedním z možných řešení (které bylo nakonec použito) bylo využití pořadí dokumentu, které vrací každý server. V přehledu nalezených dokumentů tak přibyl další sloupeček informující o celkovém pořadí. Toto Výsledné pořadí (a stejně tak skóre) se vypočte jako aritmetický průměr pořadí (skóre) dosažených na jednotlivých serverech. Způsob, jak tento problém řeší ostatní metavyhledávače, se bohužel nepodařilo zjistit (jejich výrobci odmítli podat bližší popis z důvodu utajení těchto algoritmů před konkurencí).

## 5.4 Aplikace

Pro zobrazování nalezených dokumentů bylo uvažováno o několika alternativách:

- Zobrazení pomocí externího webového prohlížeče – nešlo by prohlížet dokumenty jiného typu než HTML stránky (snad jen ještě některé další formáty v závislosti na pluginech nainstalovaných do prohlížeče).
- Implementace vlastního prohlížeče v DLL knihovně – zbytečně složité, málo flexibilní. Navíc v případě, že místo grafického rozhraní by byl modul chovající se jako webová služba, by byl tento prohlížeč naprosto zbytečný.
- Zavolání externího prohlížeče, který je asociován s koncovkou příslušného dokumentu – tato možnost se nakonec ukázala jako optimální. DLL knihovna serveru, který daný dokument našel rozhoduje o jeho koncovce a také specifikuje parametry funkce `ShellExecute`, která dokument zobrazí. Není proto složité dodat zároveň s DLL knihovnou externí prohlížeč, který se bude volat.

Další problém vznikl při řešení zobrazení seznamu parametrů uživateli. Původně se uvažovalo o jednoduchém Layout manageru, který by řídil rozmístění vstupních polí pro jednotlivé parametry v dialogu. Toto se záhy ukázalo jako velmi složité a nepřehledné, protože výsledný seznam může obsahovat až několik desítek parametrů (plus další nově vytvořené kopie).

V konečném řešení (Obrázek 5.9) jsou parametry reprezentovány jednoduchým stromem, který obsahuje maximálně dvě úrovně, kterými jsou jednak vlastní parametr (`TDISScalarParam`) s hodnotou a případně skupina (`TDISGroupParam`), která parametr obsahuje. Komponenty pro zadání dat skupiny parametrů jsou v dialogu umístěny jednoduše pod sebou.

Toto řešení je velmi přehledné a elegantní. Hodnoty nastavených parametrů lze vidět přímo ve stromě a dialog není přeplácáný díky tomu, že se zobrazuje pouze vstupní pole vybraného parametru (případně několik komponent, pokud je parametr součástí skupiny).

Uživatelské prostředí aplikace neobsahuje mnoho dalších zajímavých technik. Možnosti, které uživateli nabízí jsou blíže popsány v uživatelské dokumentaci. Jedinými zajímavými oblastmi je export seznamu dokumentů a formát ukládání dat (nastavení aplikace).

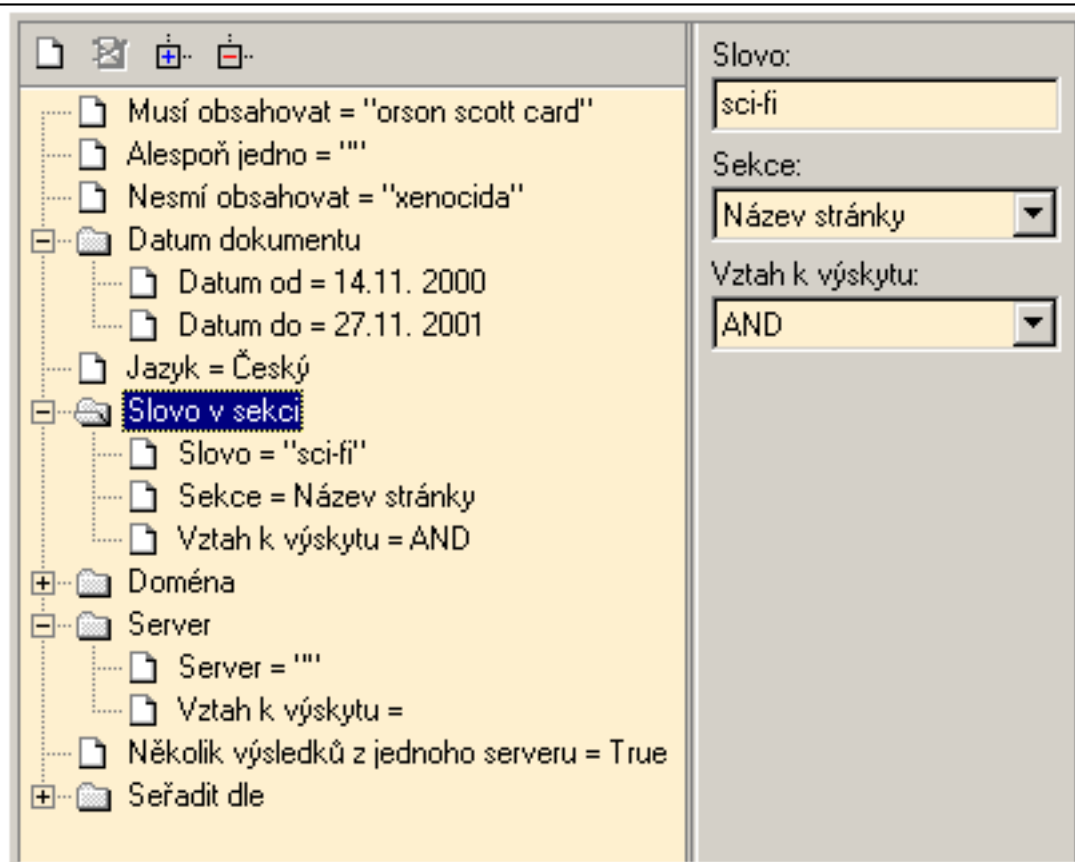
### 5.4.1 Výstupní šablony

Seznam nalezených dokumentů, který aplikace po vyhledání zobrazí je možné uložit do souboru a později znovu načíst. Další možností uchování těchto dat je export. V současné době jsou podporovány tyto formáty:

**HTML dokument (html).** Výpis vygenerovaný jako HTML stránka s použitím kaskádových stylů.

**Comma Separated Value (csv).** Tento formát lze načíst do většiny tabulkových programů jako je například Microsoft Excel (kde lze s daty dále pracovat). Jednotlivé hodnoty jsou ve výstupním souboru odděleny čárkou, věty jsou odděleny odřádkováním.





Obrázek 5.9: Strom parametrů

**Čistý text (txt).** Univerzální formát. Neobsahuje žádné formátovací tagy, výstup lze prohlížet na libovolné platformě.

Je zde také možnost přidat další šablony pro export. Stačí vytvořit příslušný textový soubor (jeho popis bude následovat dále) a uložit jej do podadresáře `Export Templates`. Na jeho jméně a koncovce nezáleží.

#### 5.4.1.1 Popis souboru šablony

Šablonu exportu tvoří jednoduchý textový soubor, který kromě vlastního formátování výstupního exportu (například HTML tagy v případě HTML exportu) obsahuje i speciální tagy, které aplikace nahradí příslušnými daty. Tyto tagy mají následující syntaxi:

```
<#JMENO_TAGU JMENO_ATRIBUTU='HODNOTA' >
```

Jména možných tagů i s vysvětlivkami viz Tabulka 5.1. Rozšiřující atributy jsou použity pouze u jediného tagu `TEMPLATE_INFO`, který popisuje šablonu a musí být uveden na samém začátku souboru.

Popis jeho atributů:

`Ext` Koncovka, s jakou se bude soubor ukládat (`html`, `txt`, `xml`, ...). Tato koncovka je poměrně důležitá, neboť to bude výchozí koncovka, pokud uživatel nezapíše jinou. Také určuje který prohlížeč se použije pro zobrazení výsledného exportu (dle nastavení asociací v systému).

`Name` Jméno (může být i s krátkým komentářem), pod jakým bude šablona uvedena v seznamu dostupných formátů pro export.

#### 5.4.2 INI soubory

Pro čitelné ukládání nastavení aplikace, historie dotazů a seznamů nalezených dokumentů bylo rozhodnuto používat nějakou formu textového souboru. Třída `TIniFile` implementovaná ve vývojovém prostředí Delphi od jeho první verze se záhy ukázala jako nedostatečná z několika důvodů:

- Omezená velikost ukládaných dat.
- Operace byly velmi pomalé.
- Pouze dvou úrovněová struktura ukládaných dat (sekce a vlastní hodnoty).

Proto byla vytvořena nová třída `TSetFile`, která všechny výše uvedené nedostatky plně odstraňuje: není omezena velikostí dat, operace jsou podstatně rychlejší (za pomoci optimalizace) a umožňuje ukládat tří-úrovněovou strukturu dat (sekce, podsekce, vlastní hodnota).

Třída `TSetFile` umožňuje načítat a ukládat proměnné typu `Integer`, `Boolean`, `Extended`, `TDateTime` a `String` (pro řetězec je určena ještě jedna metoda na ukládání „dlouhého“ řetězce, který může obsahovat odřádkování)

Název tagu	Význam
ACT_DATE	Aktuální datum a čas (tedy datum a čas exportu)
USER_NAME	Jméno aktuálně přihlášeného uživatele
DOC_COUNT	Počet nalezených dokumentů
QNAME	Název dotazu
QCREATED	Datum a čas vytvoření dotazu
QEDITED	Datum a čas poslední modifikace dotazu
QSEARCHED	Datum a čas posledního vyhledávání dokumentů odpovídajících dotazu
QTOTAL_COUNT	Maximální počet nalezených dokumentů z jednoho serveru
QSERVER_COUNT	Počet dotazem využívaných serverů
QSERVERS	Seznam serverů, které dotaz používá pro vyhledávání
QNOTE	Poznámka k dotazu
DNUM	Aktuální číslo dokumentu
DNAME	Název dokumentu
DLOCATION	Lokace dokumentu
DSCORE	Celkové skóre dosažené dokumentem
DORDNUM	Celkové pořadí dokumentu
DCREATED	Datum vytvoření dokumentu
DMODIFIED	Datum poslední změny dokumentu
DEXTRACT	Ukázka z dokumentu
DNOTE	Poznámka k dokumentu
DSERVERINFOS	Informace o hodnocení dokumentu servery, které jej našly
TEMPLATE_INFO	Tag popisující vlastní šablonu, musí být uveden na začátku souboru (tedy jako první tag), má dva atributy.
LIST_BEGIN	Tag označující začátek smyčky, která obsahuje všechny nalezené dokumenty. Mezi tagy LIST_BEGIN a LIST_END by měly být uvedeny tagy popisující dokument, které budou nahrazeny skutečnými vlastnostmi nalezených dokumentů.
LIST_END	Konec smyčky dokumentů

Tabulka 5.1: Názvy a popisy tagů šablony

Veškerá data souboru se načtou metodou `Load` do paměti, konkrétně do objektu `TStringList`, kde se jednotlivé řádky převedou na vnitřní strukturu, která je optimálnější pro další práci. To spočívá mimo jiné ve zrušení uvozujících mezer a vypuštění přebytečných dat (prázdné řádky a komentáře). V každém řádku vnitřní struktury je na začátku uvedeno jméno hlavní sekce a podsekce, poté následuje jméno atributu a vlastní hodnota.

Příklad převodu dat na vnitřní strukturu:

Data v uloženém souboru:

```
; -----
; Toto je INI soubor uzivatele
; -----
[[Application]]
  [Window]
    Left=50
    Top=62
    Width=923
    Height=644

[[Global]]
  [DuplServers]
    ServerCount=1
    ServerClass0=TDISOracleServer
```

Data ve vnitřní struktuře:

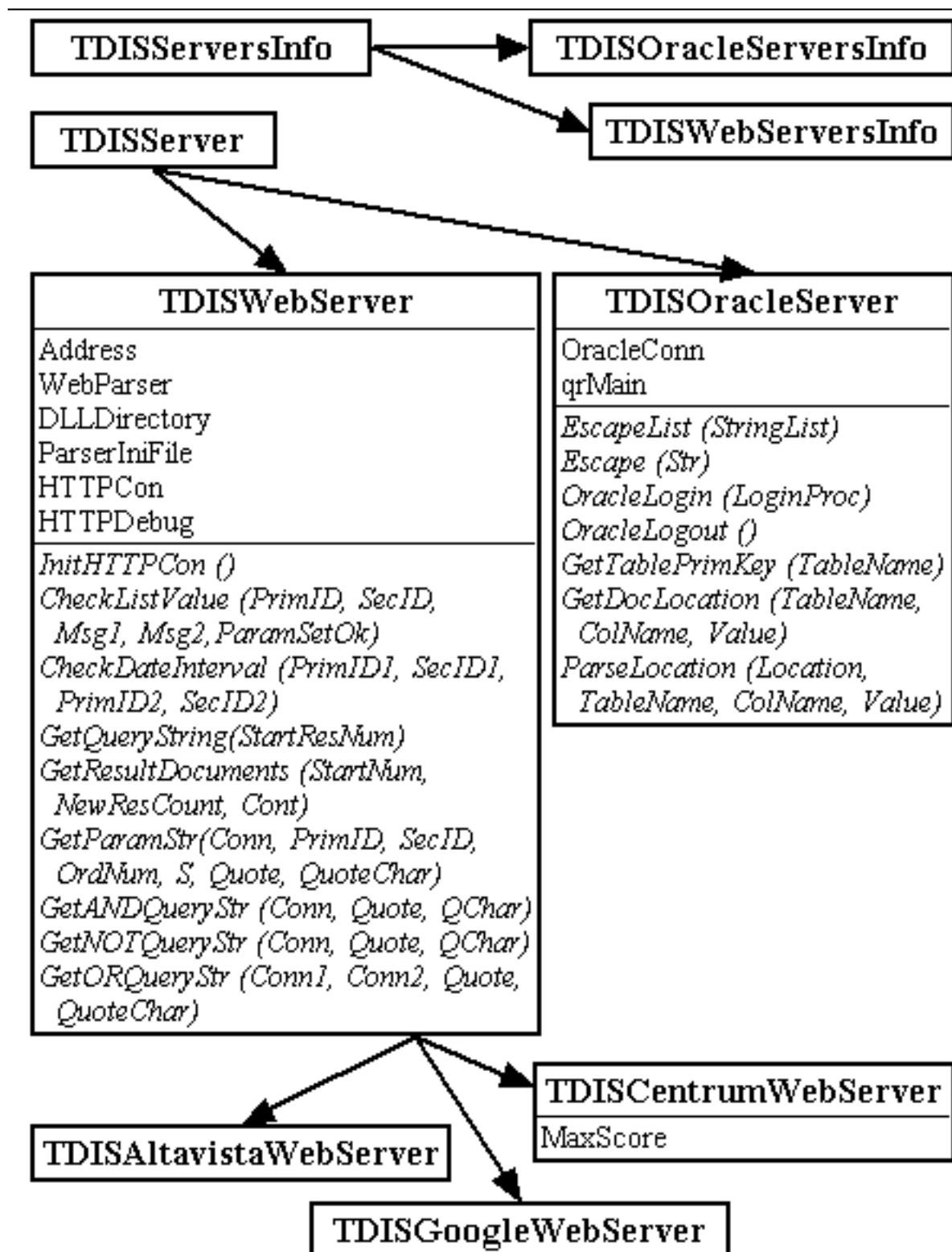
```
[Application][Window]
[Application][Window]Left=50
[Application][Window]Top=62
[Application][Window]Width=923
[Application][Window]Height=644
[Global][DuplServers]ServerCount=1
[Global][DuplServers]ServerClass0=TDISOracleServer
```

Jelikož se tento formát používá převážně pro ukládání nastavení, je vysoce pravděpodobné, že data se budou číst v tom pořadí, v jakém jsou uložena. Proto zde byla zavedena optimalizace spočívající v zapamatování poslední pozice, kde se data četla. Při dalším požadavku se data hledají od této pozice dále, a teprve v případě neúspěchu se začne hledat od začátku.

## 5.5 Implementace jednotlivých serverů

### 5.5.1 Implementace webových vyhledávačů

Pro realizaci webových fulltextových vyhledávačů byl vytvořen `TDISWebServer`, ze kterého se odvozují objekty pro konkrétní servery (`TDISAltavistaWebServer`, `TDISCentrumWebServer` a `TDISGoogleWebServer`).



Obrázek 5.10: Implementace serverů

Hlavním úkolem tohoto objektu je komunikace s příslušným WWW serverem přes protokol HTTP (HyperText Transfer Protocol). Tuto komunikaci zajišťují komponenty `TTidHTTP` a `TTidLogDebug` vývojového nástroje Delphi 6.

Jelikož `TDISWebServer` je jen meziobjekt, nechává většinu důležitých metod prázdnou, implementuje pouze společnou část pro všechny webové vyhledávače:

- Komunikace se serverem.
- Získání (stažení) stránky ze serveru (`SaveDoc`).
- Vytvoření některých základních společných parametrů pro nastavení serveru (nastavení proxy) a pro dotaz.
- Kontroly společných parametrů (`QueryCheck`).
- Metodu pro vlastní vyhledávání a získávání adekvátních dokumentů.
- Pomocné procedury pro práci s často používanými parametry.

Ostatní metody musí být implementovány až u jednotlivých potomků:

- Vytvoření parametrů specifických pro daný server.
- Kontrola těchto specifických parametrů.
- Vytvoření dotazového řetězce (tedy poskládání URL adresy směřující na webový vyhledávač a obsahující všechny příslušné parametry dotazu).
- Vypreparování vlastních hodnot z vrácené HTML stránky za pomoci objektu `TTextParser`.

### 5.5.1.1 Parser

Při využívání webových fulltextových vyhledávačů vznikl problém, jak získat informace, které jsou obsaženy v odpovědní stránce mezi dalším balastem, jež tato stránka obsahuje. Problém byl vyřešen textovým parserem `TTextParser` s externím konfiguračním souborem, který umožňuje snadno a pružně reagovat na změnu struktury stránky s výsledky (postačí změnit několik řetězců).

Vlastní objekt `TTextParser` obsahuje pole objektů typu `TTokenGroup`. Každá skupina (`TTokenGroup`) se skládá z několika (samozřejmě může obsahovat i jen jeden samotný token) tokenů (`TToken`). Token je nejmenší jednotka, představující hledanou informaci v textovém řetězci. Skládá se z těchto položek (Obrázek 5.11):

**Name** Jméno (identifikátor) tokenu, důležité při závěrečném získávání dat z parseru.

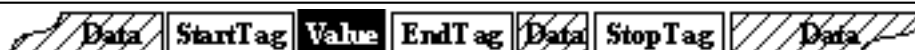
**Value** Vlastní hodnota (údaj, který je třeba získat ze vstupního řetězce).

**Mandatory** Povinnost tokenu, využije se při parsování celé skupiny. Jestliže skupina po naporsování nezískala hodnotu tokenu, který má nastaven `Mandatory=True`, pak celá skupina při získávání dat selhala.

**StartTag** Řetězec, který uvozuje vlastní hodnotu `Value`, tedy řetězec ve vstupním textu, za kterým následují důležitá data, která je třeba získat.

**EndTag** Opak **StartTagu** – řetězec, který se nachází za důležitými daty a označuje tak jejich konec.

**StopTag** Pokud se při parsování textu za vlastními daty nenalezne **EndTag**, obsahoval by token nesmyslná data. Případně by se **EndTag** mohl najít někde dále v těle dokumentu. **StopTag** slouží k vymezení oblasti hledání **EndTagu**. Pokud na něj Parser narazí (při načítání dat, tedy až poté co našel **StartTag**), znamená to, že hodnota tokenu nebyla nalezena.



Obrázek 5.11: Grafické znázornění prvků tokenu v textu

Příklad tokenu pro získání počtu Altavistou nalezených stránek:

```
TokenName="TotalFound"
StartTag="Inclusion\">We found "
EndTag=" results:</b></a><br>"
StopTag="<br>"
```

Skupina Tokenů (**TTokenGroup**) je určena pro načítání množiny dat, která se několikrát v textu opakuje. Stránka, kterou vrátí webový vyhledávač, obsahuje několik jedinečných údajů (například počet všech stránek odpovídajících položenému dotazu) a dále množinu údajů, jejichž struktura se několikrát opakuje (například informace o nalezených stránkách – název, adresa, dosažené skóre, ukázka textu, datum). Některé požadované údaje z této množiny jsou nezbytné (URL stránky), jiné jsou nepovinné (ukázka, datum a podobně). Pokud je třeba neparsovat jedinečný údaj (tento je samozřejmě definován pomocí jednoho tokenu), je pro snazší a jednodušší manipulaci použita skupina obsahující tento jediný token.

Atribut **ValueParsed** skupiny určuje, zda se povedlo při parsování načíst hodnotu. Tedy zda se načetly hodnoty pro všechny její tokeny s atributem **Mandatory = True**.

**RepeatCount** udává, kolikrát se tato skupina tokenů v parsovaném textu opakuje (a tedy kolikrát po sobě se budou její hodnoty načítat). Defaultně je **RepeatCount=0**, což znamená, že se skupina nebude opakovaně načítat.

S **RepeatCount** souvisí další atribut **SeqNum**, který určuje, o kolikátou instanci množiny jde.

Hlavní a nejdůležitější metodou je

```
Parse (S: string; StartPos: integer): integer;
```

kteřá projíždí řetězec **S** od pozice **StartPos** a snaží se v něm najít hranice a obsah tokenů skupiny. K tomu využívá jednodušší metodu

```
FindToken(S: string; var T: PToken; Start: integer): integer;
```

pro vyhledání tokenů skupiny.

Postup je jednoduchý – projíždí pole tokenů a hledá je v zadaném textu. Uspěje-li při hledání tokenu metodou `FindToken`, pokračuje hledáním dalšího od pozice, kterou tato metoda vrátila. Neuspěje-li, hledá další token od stejné pozice (pokud nenalezený token nebyl povinný), nebo skončí s nezdarem (byl-li hledaný token povinný).

Objekt `TParser` obsahuje pole skupin. Definice těchto skupin je buď obsažena přímo v kódu a vytvoří se pomocí příslušných pomocných metod (`Add`, `AddGroup`, `InsertGroup`) a nebo se jejich struktura načte z textového souboru.

V souboru lze používat řádkové komentáře (veškerý text za znakem „%“), pro větší přehlednost mohou být jednotlivé řádky indentovány a prokládány prázdnými řádky. Každý jednotlivý údaj musí být zapsán na samostatném řádku. Hodnoty tagů (`TOKENNAME`, `STARTTAG`, `ENDTAG`, `STOPTAG`) nesmí volně obsahovat znak „““ (uvozovky), byl by totiž chápán jako konec řetězce. Proto musí být escapován (uvozen znakem „\“). Stejně tak musí být ošetřen i samotný znak „\“. Má-li tedy tag obsahovat „\“, musí to být zapsáno takto: „\\“.

Jelikož formát HTML není citlivý na počet mezer či odřádkování ve zdrojovém textu stránky, není třeba tyto znaky (dokonce je zakázáno zapsat do tagu odřádkování, mezery jsou povoleny) do obsahu tagu zapisovat. Při porovnávání jsou veškeré bílé znaky (mezery, odřádkování, tabulátor) ve vstupním textu i v tagu ignorovány.

Soubor obsahuje následující klíčová slova:

`TOKEN_BEG`, `TOKEN_END` Klíčová slova označující začátek a konec tokenu.

`GROUP_BEG`, `GROUP_END` Klíčová slova označující začátek a konec skupiny, skupina obsahuje definice tokenů.

`TOKENNAME`, `STARTTAG`, `ENDTAG`, `STOPTAG`, `MANDATORY` a `REPEATCOUNT` Vlastnosti skupiny a tokenů. Další informace viz předchozí popis objektů.

Příklad ini souboru pro parser Altavisty:

```
;-----
; ParserSettings pro Altavistu
; Petr Vaclavek 4. 11. 2001
;-----

TOKEN_BEG
  TokenName="Hledane slovo"
  StartTag="- Web Results for: "
  EndTag="</title>"
  Mandatory=0
TOKEN_END

; Celkovy pocet nalezenych stranek
TOKEN_BEG
```



```

TokenName="TotalFound"
StartTag="Inclusion\">We found" % <-- znak " je escapovan!
EndTag=" results:</b></a><br>"
Mandatory=1
TOKEN_END

; Jednotlive stranky a informace o~nich
GROUP_BEG
RepeatCount=1000

TOKEN_BEG
TokenName="DocURL"
StartTag="onMouseOver=\"status=' "
EndTag=" ' ; return"
Mandatory=1
TOKEN_END

TOKEN_BEG
TokenName="DocName"
StartTag="true;\">"
EndTag=" </a>"
Mandatory=0
TOKEN_END

TOKEN_BEG
TokenName="DocExtract"
StartTag="<br>"
EndTag="<br><span"
StopTag="<span"
Mandatory=0
TOKEN_END

GROUP_END

```

Po vytvoření pole skupin (at' už pomocí volání metod nebo načtením struktury ze souboru) je třeba naparsovat vstupní text. K tomu je určena hlavní metoda objektu `TTextParser`:

```
ParseText (S: string): boolean;
```

Tato metoda prochází polem skupin a pokouší se je naparsovat. Uspěje-li a má-li se skupina opakovat (atribut `RepeatCount > 1`), vytvoří její kopii s nižším `RepeatCount` a vyšším pořadovým číslem `SeqNum` a opět se jí pokusí naparsovat. Neuspěje-li, přejde na následující skupinu.

Ke zjištění naparsovaných hodnot slouží metoda

```
GetTokenValue (TokName: string; SNum: integer;  
    var Val: string): TTokenState;
```

Hledaná hodnota je identifikována názvem tokenu a pořadovým číslem skupiny. Návrátová hodnota metody udává, zda byl token nalezen a naparsován, či nikoliv.

Před dalším parsováním je třeba odstranit kopie opakovaných skupin, jež byly vytvořené minulým prohledáváním. K tomu je určena metoda `ClearTokens`.

## 5.5.2 Implementace Oracle8i interMedia

DLL knihovna implementující Oracle8i interMedia je jediná, která využívá možnosti vytvářet více kopií serveru. Díky tomu umožňuje uživateli paralelně vyhledávat stejný dotaz na několika Oracle serverech.

Parametry, ve kterých se tyto instance odlišují, se týkají zejména nastavení (zdroj dat, jméno, heslo, název serveru) a také dotazu (jméno tabulky, ve které se bude hledat, mapování sloupců a další podmínky na atributy tabulky). Každý nově vytvořený server musí mít jiné (jednoznačné) ID těchto parametrů, proto se při vytváření serveru využije informace o čísle instance serveru (`ServerNum`) a původní ID parametru se posune o příslušnou konstantu. Navíc jsou jednotlivé servery jednoznačně identifikovány pomocí `IDName`.

Komunikaci mezi DLL knihovnou a vlastním Oracle serverem zajišťují komponenty `TADOConnection` a `TADOQuery`.

Na rozdíl od ostatních implementovaných serverů se v tomto případě plně využívá metoda `Login`. Slouží k vlastnímu připojení k serveru (pomocí `TADOConnection`) a také k získání důležitých informací o datech zde uložených. Konkrétně se jedná o získání seznamu použitelných tabulek, nad kterými byl vytvořen textový index. K těmto tabulkám ještě zjistí seznam jejich atributů. Všechny uvedené údaje vyplní do příslušných parametrů dotazu.

Vlastní vyhledávání spočívá ve vytvoření klasického SQL dotazu, který je zaslán na server a zpracování množiny záznamů odpovídajících dotazu. Dokumenty (texty, které jsou prohledávány) musí být obsaženy v tabulkách, nad kterými jsou vytvořeny textové indexy. Navíc je v aplikaci podporován pouze jeden způsob uložení textových dat, a to přímo ve sloupečku tabulky (jiné tabulky se v parametru pro výběr zdroje dat ani neobjeví). To znamená, že při vytváření textového indexu nad tabulkou byl nastaven parametr `DATASTORE` na `DIRECT_DATASTORE`. Další informace o Oracle8i interMedia viz [11].

# Kapitola 6

## Závěr

### 6.1 Srovnání s konkurenčními produkty

Jelikož DISClient využívá pro vlastní hledání dokumentů na Internetu fulltextové vyhledávače, měl by uživateli nabídnout stejné možnosti. To se týče jednak parametrů dotazu a jednak možností prostředí. Parametry lze snadno díky zavedenému systému implementovat. S prostředím je to horší, neboť běžný uživatel je zvyklý na používání oblíbeného internetového prohlížeče nejen pro vyhledávání, ale i pro prohlížení nalezených dokumentů. Je pro něj tedy nezvyklé zadávat dotazy v jedné aplikaci a výsledky si prohlížet v druhé.

Jednou z možností, jak alespoň částečně simulovat uživateli prostředí na jaké je zvyklý, je zapnutí automatického exportu (Nastavení/Export/Automaticky zobrazovat). Výsledky se po ukončení vyhledávání exportují do zvoleného formátu (například HTML stránky) a zobrazí se v nastaveném prohlížeči, kde si je může uživatel libovolně prohlížet a procházet mezi nimi.

Jinou možností je implementace dalšího modulu využívající jádro aplikace. Aplikace by se pak chovala jako další internetový metavyhledávač (viz [Oddíl 5.1](#) a také [Obrázek 5.2](#)).

Program však přináší spoustu dalších výhod a možností: ukládá historii dotazů, takže se k nim může uživatel kdykoliv vrátit, znovu vyhledat, či přímo stáhnout dokument z dříve uloženého seznamu. Po vyhledání lze s výsledky dále pracovat – uložit seznam, třídít dle různých kritérií či vyexportovat do libovolného (i vlastního) textového formátu.

Navíc uživatel může paralelně vyhledávat v několika zdrojích dat (i když to webové metavyhledávače umožňují také) a tyto zdroje nemusí být pouze fulltexty na Internetu (to už dostupné metavyhledávače neumí). Díky paralelnímu zpracování je vyhledávání podstatně rychlejší, než postupné ruční procházení jednotlivých serverů a zadávání stále stejného dotazu.

Ve srovnání s dostupnými desktopovými vyhledávači (viz [Kapitola 4, \*Obdobné aplikace\*](#)) patří mezi základní přínosy zastoupení českých vyhledávačů, cena (jedná se o volně šiřitelný program), dostupnost zdrojových kódů a hlavně rozšiřitelnost. Kdokoliv může vytvořit novou knihovnu pro vyhledávání nejen pomocí dalších webových služeb, ale i pro jakékoliv jiné zdroje dat (soubory na disku, data emailových klientů, data v databázích, knihovní systémy, ...). Navíc uživatel smí využívat téměř veškeré možnosti, které nabízí jednotlivé vyhledávací servery, plně ovlivnit způsob exportu výsledků a podobně.

Na druhou stranu ale DISClient nedisponuje některými funkcemi, které uvedené aplikace obsahují: integrace do systému, prohlížeče, vlastní prohlížeč s vestavěnými vylepšeními (např. zvýraznění hledaných slov), agenti na automatické vyhledávání nových dokumentů odpovídajících dotazu, kontrola dokumentu (dosažitelnost, vlastní explicitní kontrola, zda opravdu vyhovuje dotazu) a další. Ovšem není problém výše uvedené chybějící vlastnosti do aplikace doprogramovat.

## 6.2 Další vývoj

Aplikaci lze samozřejmě dále vyvíjet a vytvořit tak plnohodnotný produkt dosahující (a v mnohých ohledech přesahující) možnosti komerčních aplikací.

Hlavní možnosti rozšíření aplikace:

- Vytvoření nových DLL knihoven pro další servery:
  - Další webové fulltextové vyhledávače (Atlas, Excite, MSN, . . .).
  - Jiné webové služby, například hledání v diskusních skupinách (Google Groups, Serge), vyhledávání ve slovnících, encyklopediích (Britannica) a další.
  - Vyhledávání dokumentů v dalších databázích (MS SQL, Interbase, Sybase, Informix a podobně).
  - Úplně nové oblasti – fulltextové prohledávání souborů na lokálním disku, korespondence v emailových klientech, . . .
- Vylepšení stávajících DLL knihoven: přidání dalších parametrů zejména pro Oracle či interMedia (například zapojení slovníku, prohledávání v datech, které nejsou uloženy přímo ve sloupečku tabulky), využití koláčků (cookie) pro nastavení webových vyhledávačů (hlavním přínosem by bylo získání více informací o nalezených dokumentech, neboť tyto vlastnosti je třeba nastavit právě pomocí cookie).
- Místo GUI vytvořit jiný modul, jenž by využíval hledacího jádra aplikace (DISEngine) viz Obrázek 5.2.
- Mnoho dalších drobných vylepšení uživatelského prostředí, kterými disponují konkurenční produkty (zejména jejich placené „Plus“ a „Pro“ verze).

## 6.3 Zhodnocení a závěr

Výsledkem diplomové práce nazvané „Klient DIS“ je plnohodnotná aplikace, která umožňuje uživateli vyhledávat v několika různorodých datových zdrojích.

Velkou výhodou je skutečnost, že ačkoliv je program hotový, lze jej dále rozšiřovat, co do rozsahu implementovaných funkcí v jednotlivých knihovnách DLL, tak do množství dalších knihoven, které si díky univerzálnímu rozhraní může každý vývoják napsat sám. Dalším velkým kladem je oddělení vlastního jádra od uživatelského prostředí, není proto problém vytvořit zcela jinou aplikaci (službu) využívající toto „multivyhledávací“ jádro.

Jeden z hlavních přínosů je řešení systému parametrů, který umožňuje vývojářům implementovat téměř libovolný parametr dotazu nového DIS Serveru. Navíc bylo poměrně elegantně vyřešeno zobrazení těchto parametrů uživateli.

Při vývoji byla snaha pokrýt všechny nedostatky zjištěné u obdobných nástrojů, kterými byl vývoj vlastní aplikace značně inspirován, což se z velké části podařilo.

# Literatura

- [1] Jaroslav Pokorný: *Výběr informací v textových bázích dat*, OVC ČVUT, srpen 1989.
- [2] Jaroslav Pokorný, Václav Snášel, and Dušan Húsek: *Dokumentografické informační systémy*, Karolinum, 1998.
- [3] Petr Václavek: *Klientské vyhledávače*, Softwarové noviny 8/2001, strana 74-77.
- [4] Petr Václavek: *Altavista*, Softwarové noviny 11/2001, strana 64-67.
- [5] Petr Václavek: *Fulltextové vyhledávače*, Softwarové noviny 12/2001, strana 80-82.
- [6] *The Web Robots Pages*, <http://www.robotstxt.org/wc/robots.html>.
- [7] *AltaVista*, <http://www.altavista.com/sites/help>.
- [8] *Google*, <http://www.google.com/about.html>.
- [9] *fulltext.Centrum*, <http://fulltext.centrum.cz/index.php>.
- [10] *Megatext*, <http://www.megatext.cz/popis.htm>.
- [11] *Oracle8i Online Generic Documentation Library*,  
<http://mates.ms.mff.cuni.cz/oracle/doc/ora815nt/>.
- [12] *Open Directory*, <http://dmoz.org/help/helpmain.html>.

# Příloha A

## Uživatelská dokumentace

### A.1 Předmluva

DISClient je nástroj sloužící pro snadné fulltextové vyhledávání nejen pomocí známých webových vyhledávačů jako je například Altavista, Google a podobně, ale umožňuje také přidávat další zdroje dat, které mohou obsahovat hledané dokumenty (například data uložená na Oracle8i interMedia).

Hlavní výhodou tohoto systému je počet oslovených serverů a přívětivé prostředí, ve kterém si uživatel definuje svůj dotaz. Díky tomu se nemusí učit speciality jednotlivých vyhledávacích serverů a nemusí znát syntaxi na nich pokládaných dotazů. Práce je tak podstatně snazší, rychlejší a efektivnější.

Další výhodou je možnost nastavení prostředí, přednastavitelné speciální funkce jako je například rychlé hledání, či export. Nezanedbatelné je také uchovávání historie hledání a možnost uložení a exportu výsledků. Kdykoliv se tak může uživatel vrátit ke staršímu dotazu a znovu vyhledat odpovídající dokumenty nebo jen načíst informace o dříve nalezených dokumentech.

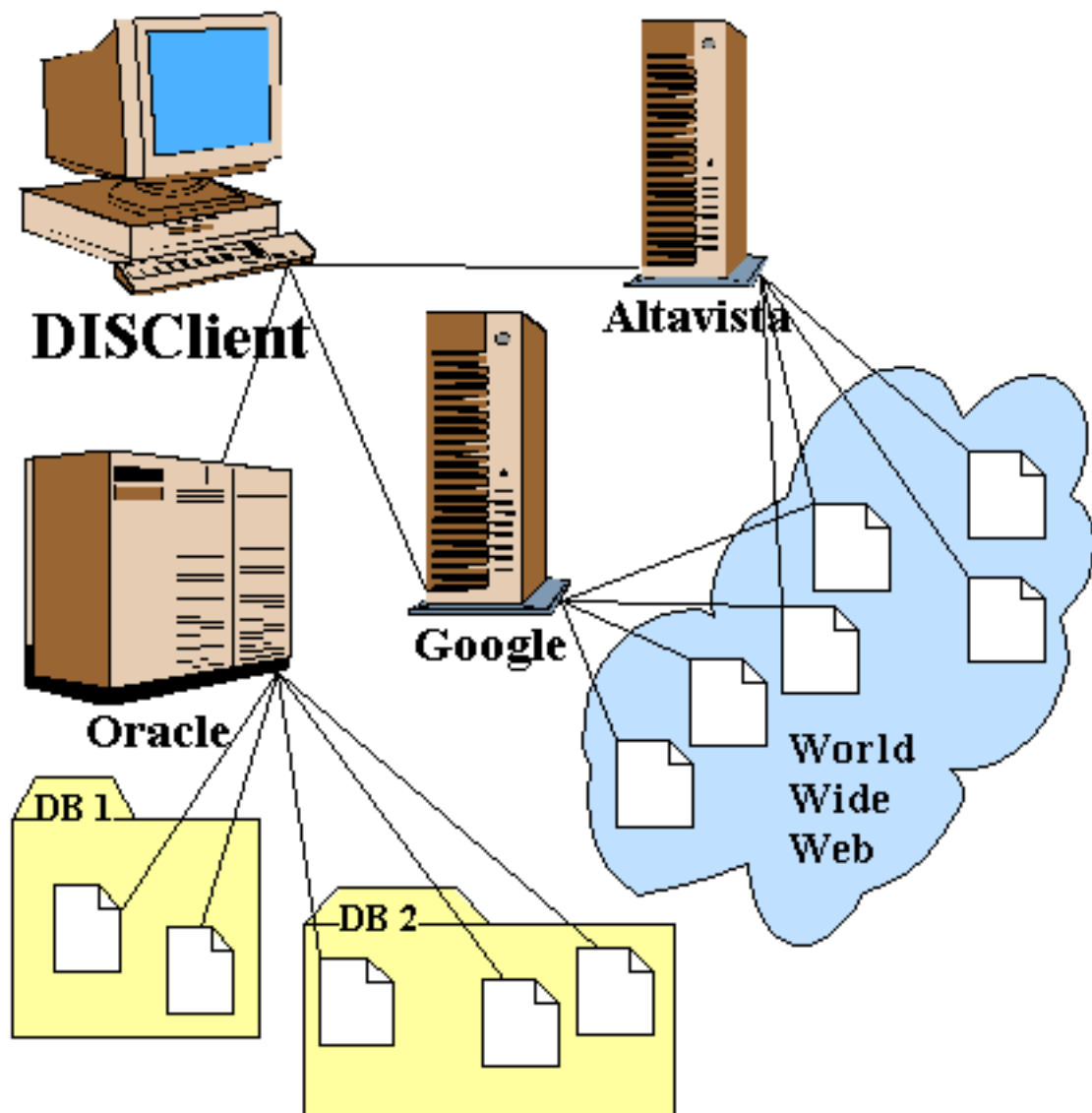
Díky tomu, že aplikace využívá více zdrojů dat, zpracuje podstatně více (a kvalitněji) jejich výsledky než by to provedl samotný uživatel postupným použitím jednotlivých serverů. Aplikace tyto servery oslovuje paralelně a navíc automaticky filtruje duplicity a umožňuje uspořádat nalezené dokumenty.

Veškerý popis práce s aplikací DISClient je náplní tohoto manuálu. Pro pochopení je potřeba znát základní pojmy a mít jisté znalosti při práci v prostředí Microsoft Windows.

### A.2 Požadavky

#### A.2.1 Minimální konfigurace

- Procesor kompatibilní s procesorem Pentium 300 MHz.
- 5 MB volného místa na pevném disku.
- Operační systém Microsoft Windows 2000.
- V případě používání knihovny pro Oracle je potřeba mít nakonfigurován ODBC ovladač pro Oracle 8, odpovídajícího klienta (ten je součástí klientské instalace



Obrázek A.1: Schéma práce aplikace DISClient



Oracle) a podpora ADO (Microsoft ActiveX Data Objects) verze 2.1 (nebo vyšší). Ta je již součástí operačních systémů Microsoft Windows 2000 a XP. Pro jiné systémy ji lze zdarma stáhnout na stránkách Microsoftu (<http://www.microsoft.com>).

- Internetové připojení.

## A.2.2 Doporučená konfigurace

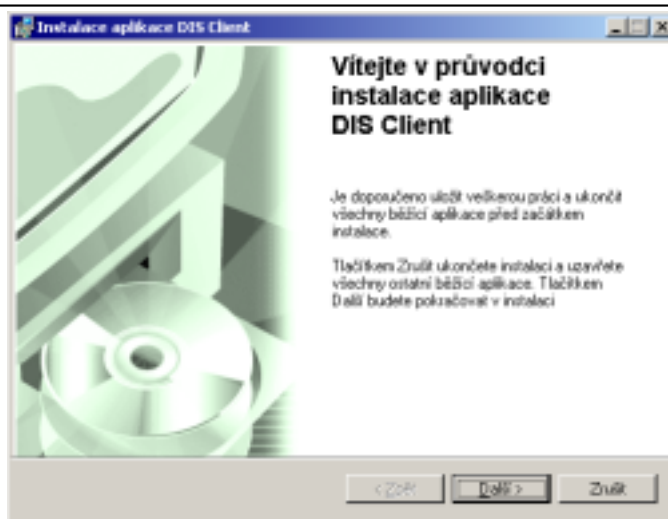
- Procesor kompatibilní s procesorem Pentium III 500 MHz.
- Polohovací zařízení (myš, tablet, ...).
- Aplikace pro prohlížení nalezených dokumentů a vygenerovaných exportů. A to zejména prohlížeč HTML stránek.

## A.2.3 Poznámky ke starším operačním systémům

Aplikace byla koncipována tak, aby běžela na operačním systému Windows 2000 hlavně z důvodu chyb v dřívějších operačních systémech. Zejména se jedná o velmi časté chyby, jež obsahuje knihovna `comctr132.dll`, kterými trpí zejména produkty vytvořené v Delphi či C++ Builderu.

V žádném případě tím ale není řečeno, že by produkt na jiné platformě než Windows 2000 neběžel, ale pouze to, že tam nemusí pracovat tak, jak bylo zamýšleno právě díky výše zmíněným problémům.

## A.3 Instalace

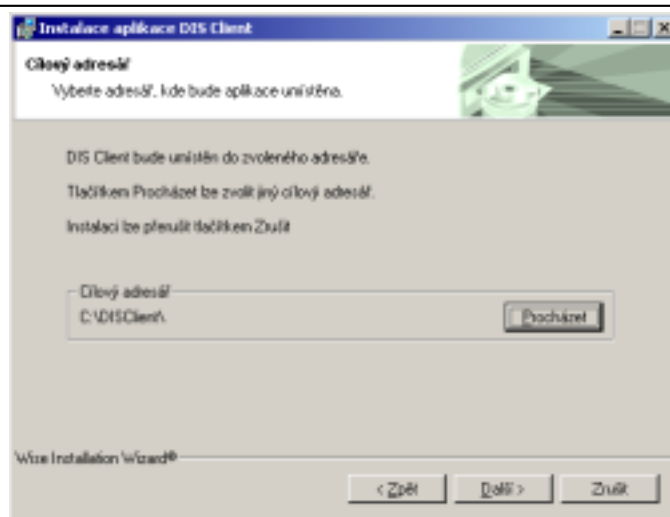


Obrázek A.2: Úvodní dialog

Instalace aplikace DIS Client je velmi jednoduchá a probíhá v několika málo krocích. Lze ji kdykoliv zastavit tlačítkem Zrušit. Uživatel pak bude dotázán, zda opravdu chce instalaci ukončit. K dokončení se pak může kdykoliv vrátit.

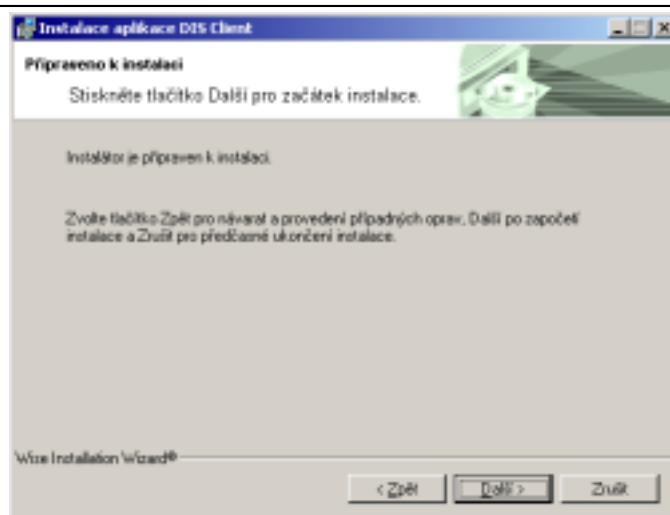
Prvním krokem je spuštění instalačního souboru `DISClient.msi`. Po inicializaci se objeví úvodní dialog (Obrázek A.2).

Po stisku tlačítka **Další** se zobrazí dialog pro zadání lokace na disku, kam se má aplikace nainstalovat (Obrázek A.3). Změna cílového adresáře se provede za pomoci tlačítka **Procházet**.



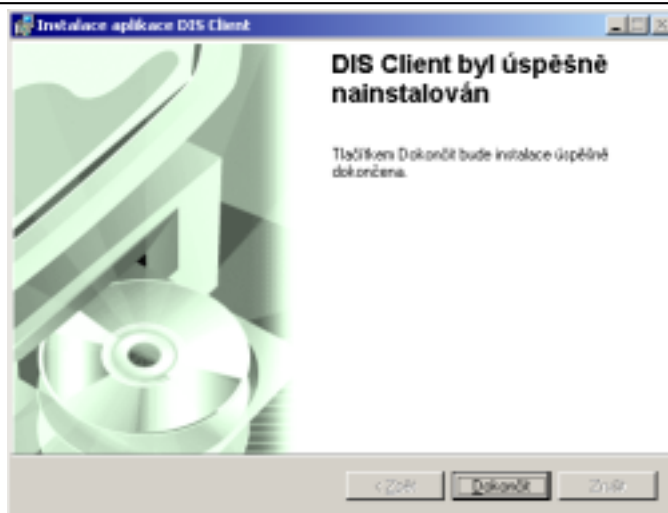
Obrázek A.3: Výběr adresáře

Nyní je instalátor připraven nainstalovat DIS Client na uživatelský disk. (Obrázek A.4). To spočívá v nakopírování souborů do zadaného adresáře, vytvoření zástupce v nabídce Start a přidání programu do seznamu nainstalovaných aplikací (Start/Nastavení/Ovládací panely/Přidat nebo odebrat programy), odkud lze kdykoliv snadno odinstalovat.



Obrázek A.4: Instalace připravena

Po úspěšné instalaci se objeví informační dialog (Obrázek A.5) a v hlavní nabídce systému START/Programy vznikne zástupce pro spuštění aplikace.



Obrázek A.5: Instalace úspěšně dokončena

Nechce-li uživatel instalovat aplikaci standardním způsobem, postačí když z instalačního CD zkopíruje adresář DISClient na pevný disk a spuštění poté provede souborem DISClient.exe.

Odinstalování aplikace se provede klasickým způsobem – použitím volby Přidat nebo odebrat programy z Ovládacích panelů.

Instalační soubor je vytvořen pomocí nové technologie „Windows Installer“. Místo toho, aby každý produkt měl vlastní instalační program, je instalace prováděna pomocí tzv. instalační databáze (soubor s příponou .MSI). Tato databáze obsahuje informace o tom, kam se má produkt instalovat, jaké přípony se mají registrovat a další.

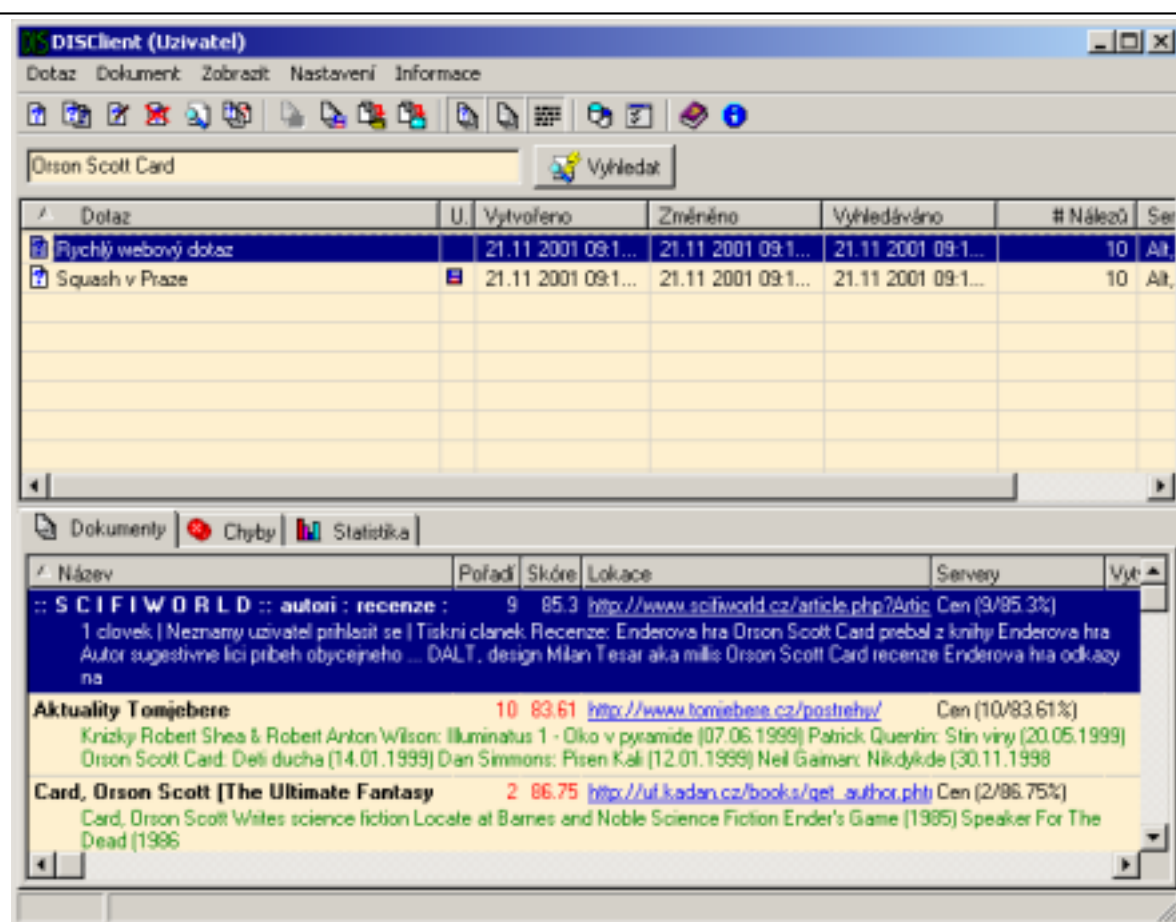
V případě instalace produktu na starší operační systém, kde ještě tato technologie není zavedena, je potřeba ji dodatečně nainstalovat (potřebný instalátor lze najít na stránkách Microsoftu případně na instalačním CD v adresáři Servis, soubor InstMsi.exe).

## A.4 Práce s aplikací

### A.4.1 Popis aplikace

Hlavní okno aplikace (Obrázek A.6) se skládá z několika částí: hlavní nabídka, nachází se v horní části okna. Hned pod ní je panel nástrojů, který obsahuje nejčastěji používané akce z hlavní nabídky. Panel pro rychlé vyhledávání se vstupním políčkem pro zadání hledaného výrazu a tlačítkem pro vlastní vyhledávání.

Největší část okna zabírá přehled dotazů a seznam nalezených dokumentů, který se váže k označenému dotazu v přehledu. Nejspodnější část okna zabírá stavový řádek, který zobrazuje informace o aktuálně vybraném tlačítku v panelu nástrojů nebo o položce v hlavní nabídce.



Obrázek A.6: Hlavní okno aplikace

### A.4.2 Přihlášení

Prvním krokem při používání aplikace DISClient je přihlášení uživatele do systému. Různé uživatelské profily umožňují individuální nastavení pracovního prostředí, vlastní volbu serverů, historii dotazů atd.



Obrázek A.7: Dialog pro přihlášení

Při startu aplikace se objeví přihlašovací obrazovka s logem (Obrázek A.7), která nabízí seznam uživatelů. Samozřejmě je zde také možnost vytvořit zcela nového uživatele. Jiný způsob přihlášení je spuštění aplikace s parametrem, který představuje jméno uživatele:

```
DISClient.exe UŽIVATEL
```

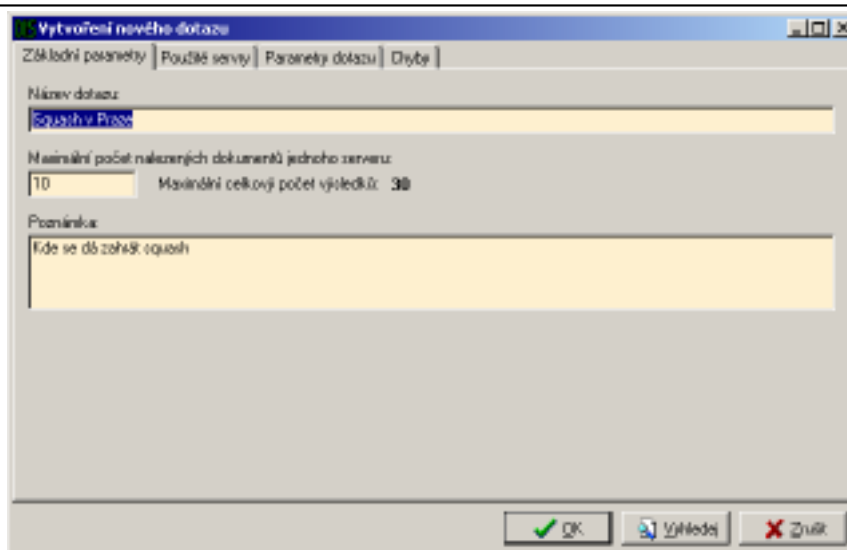
### A.4.3 Vytvoření a editace dotazu

Jelikož je hlavním posláním aplikace usnadnit vyhledávání, je nejdůležitějším prvkem vlastní dotaz.

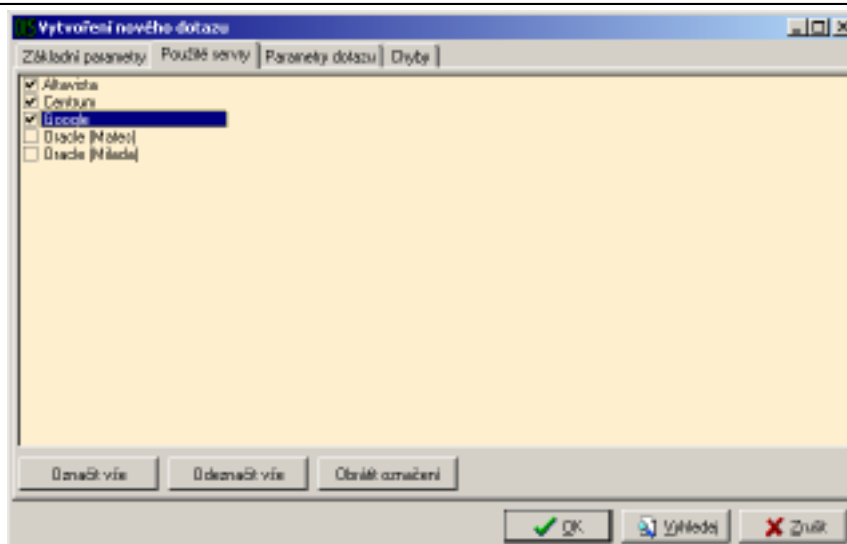
Nový dotaz se vytvoří za pomoci položky Dotaz/Nový dotaz v hlavní nabídce, případně lze použít odpovídající tlačítko v panelu nástrojů.

Prvním krokem je zadání názvu dotazu (ten slouží jen pro jeho identifikaci v seznamu dotazů). Zvolení počtu dokumentů, které si při vyhledávání aplikace vyžádá od každého osloveného serveru. A případně je možné zapsat poznámku (Obrázek A.8).

Druhým krokem je výběr serverů (Obrázek A.9), které budou zadaný dotaz vyhodnocovat. Některé servery se musí před vlastním použitím inicializovat (například

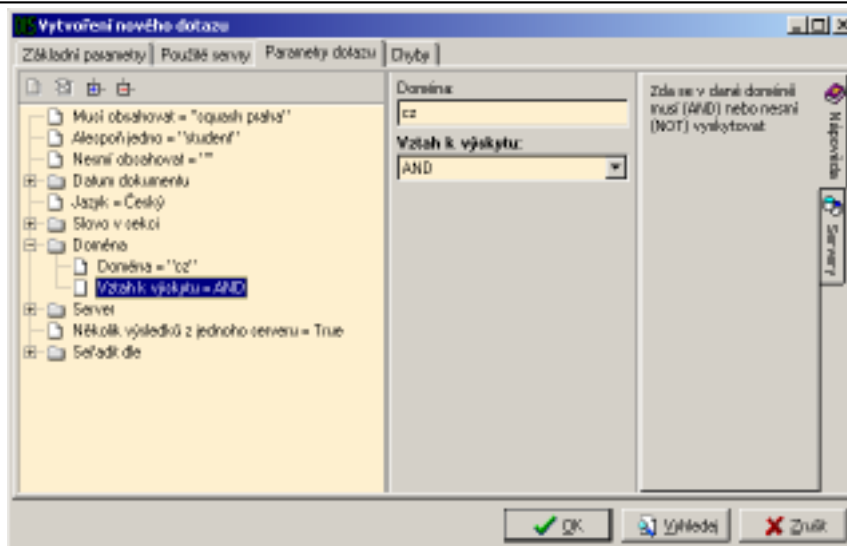


Obrázek A.8: Základní vlastnosti dotazu



Obrázek A.9: Výběr serverů

Oracle). Toto může trvat delší dobu, ale děje se tak jen při prvním využití serveru. Průběh připojování se zobrazuje ve zvláštním okně. Podle toho, které servery zde budou vybrány, bude vypadat strom parametrů dotazu v dalším kroku.



Obrázek A.10: Parametry dotazu

Na další záložce je zobrazen strom parametrů dotazu (Obrázek A.10). Dotazy jsou dvojího typu. Jednoduché představující jedinou hodnotu (například parametr „Jazyk“ pro specifikaci jazyku dokumentu) nebo skupinové. Ty reprezentují skupinu několika jednoduchých parametrů společně popisující jednu vlastnost hledaného dokumentu (například parametr „Datum dokumentu“ pro vymezení intervalu vzniku dokumentu nebo parametr „Slovo v sekci“ popisující která slova se musí případně nesmí v zadané sekci vyskytovat).

Po vybrání parametru se v prostřední části okna objeví příslušná editační políčka pro vyplnění jeho hodnoty (hodnot).

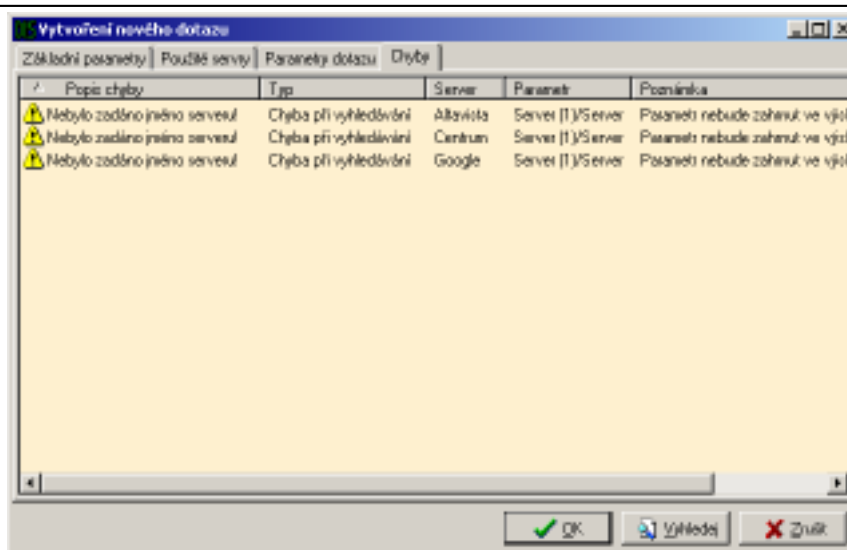
Někdy je potřebné vyplnit více parametrů stejného typu. Například při specifikaci výše zmíněného parametru „Slovo v sekci“ může uživatel potřebovat zadat jedno slovo, které se musí nacházet v určité sekci a zároveň ještě chce aby hledaný dokument neobsahoval nějaké jiné slovo v jiné sekci. Toto lze samozřejmě u některých parametrů (u kterých to má smysl) provést. Slouží k tomu jednak položka Přidej parametr v kontextové nabídce parametru, případně této položce odpovídající tlačítko nad stromem. Nově vytvořený parametr bude obsahovat stejné hodnoty jako originál. Libovolný nově vytvořený parametr, či jeho originál, lze obdobně smazat (položka Smaž parametr).

Poslední dvě tlačítka v této nástrojové liště slouží pro kompletní rozbalení případně sbalení stromové struktury parametrů.

Pravá část dialogu zobrazuje pomocné informace o aktuálně vybraném parametru. První záložka zobrazuje nápovědu, druhá přehled serverů, které daný parametr využívají. Výběr serverů se na tomto místě nedá změnit, to lze provést v dialogu pod záložkou Použité servery viz předcházející krok.

V okamžiku, kdy je uživatel spokojen s dotazem, potvrdí dialog tlačítkem Ok, případně Vyhledej (pokud dotaz neobsahuje žádnou chybu, program se pokusí ihned začít vyhledávat). V tomto okamžiku se provede kontrola uživatelem zadaných dat a

pokud se naleznou nějaké chyby, nabídne uživateli jejich zobrazení v poslední záložce tohoto dialogu (Obrázek A.11).



Obrázek A.11: Nalezené chyby dotazu

Tato poslední záložka obsahuje seznam chyb a informace o nich (popis, jméno serveru, jméno parametru, kterého se týká a případně další informace v poznámce – typicky jak bude chybný parametr zpracován v případě, že nebude opraven).

Rozhodne-li se uživatel chybu opravit, stačí na ni poklepat a dialog se automaticky přepne na předchozí záložku a nalezne tento parametr ve stromu.

Neobsahuje-li dotaz žádné chyby, je připraven pro vyhledání dokumentů, které tomuto dotazu odpovídají. Dokumenty budou hledány pomocí serverů, jež byly vybrány v druhém kroku.

Jiný způsob vytvoření nového dotazu je jeho odvození od stávajícího. Vznikne tím totožná kopie originálního dotazu. Toto je zejména výhodné, když se uživatel rozhodne ladit dotaz, ale chce si ponechat jeho kopii, aby se k němu mohl později v případě neúspěchu vrátit.

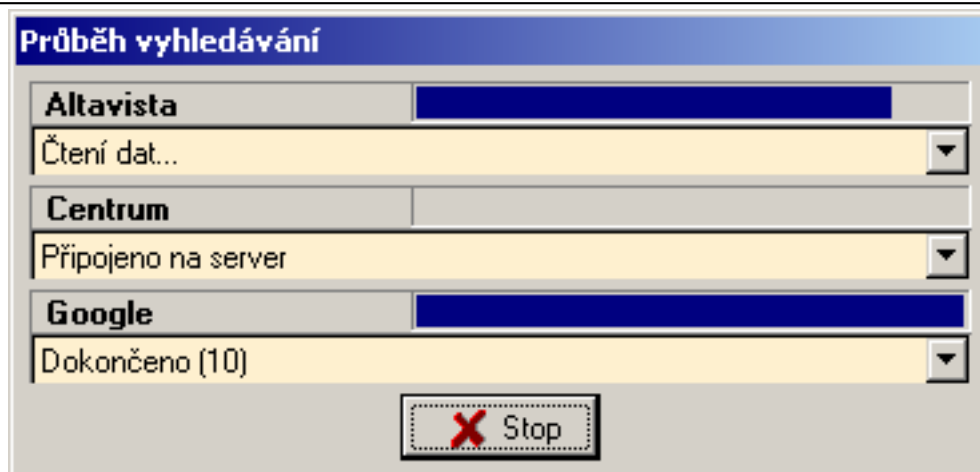
Existuje ještě jeden způsob vytvoření nového dotazu. Stačí do vstupního pole pod panelem nástrojů zadat řetězec, jež se má vyhledat a potvrdit tlačítkem Vyhledat. Parametry nově vytvářeného dotazu lze vyplnit v „šabloně“, která se nachází v nastavení. Nově vytvořený dotaz se zařadí do seznamu a ihned se začne vyhledávat.

#### A.4.4 Vyhledávání

Vyhledávání dokumentů se spustí buď za pomoci položky Dotaz/Vyhledat v hlavní nabídce případně odpovídajícím tlačítkem v nástrojové liště a nebo tlačítkem Vyhledat přímo v dialogu pro editaci dotazu. V případě, že se s některými servery ještě nepracovalo, proběhne jejich inicializace (například Oracle).

V následujícím dialogu (Obrázek A.12) se zobrazuje postup vyhledávání na jednotlivých serverech. Pod názvem serveru je vždy combobox se zprávami o průběhu. Při ukončení hledání obsahuje toto políčko mimo jiné počet nalezených dokumentů.





Obrázek A.12: Průběh vyhledávání

Hledání lze kdykoliv ukončit tlačítkem Stop. Změní-li se popiska tohoto tlačítka na Ok, znamená to, že vyhledávání bylo ukončeno.

Název	Pořadí	Skóre	Lokace	Servery
Card, Orson Scott [The Ultimate Fantasy Card, Orson Scott Writes science fiction Locate at Barnes and Noble Science Fiction Ender's Game (1985) Speaker For The Dead (1986)	3	86.75	<a href="http://wikisource.org/Book:logst_south">http://wikisource.org/Book:logst_south</a>	Can (3/86.75%)
Card, Orson Scott [The Ultimate Fantasy Card, Orson Scott Writes science fiction Locate at Barnes and Noble Science Fiction Ender's Game (1985) Speaker For The Dead (1986)	2	86.75	<a href="http://wikisource.org/Book:logst_south">http://wikisource.org/Book:logst_south</a>	Can (2/86.75%)
<a href="http://scifi.wikisource.org/5000/Anketa.php">http://scifi.wikisource.org/5000/Anketa.php</a> 1. Heinlein Robert A. 12 2. Tolkien J. R. R. 6 3. Asimov Isaac 4 3. Card Orson Scott 4 4. Clarke Arthur C. 3 4. Morozov John 3 5. Harrison Harry 2 5. Lenka Lanczova 2 5. Adams 1 5. Aldis Brian W. 1 5. Foster Alan Dean 1 5. Orson Scott Card 1 5. J. r. r. tolkien 1 Pořadí	7	85.3	<a href="http://scifi.wikisource.org/5000/Anketa.php">http://scifi.wikisource.org/5000/Anketa.php</a>	Can (7/85.3%)

Obrázek A.13: Seznam nalezených dokumentů

Nyní se ve spodní části hlavního okna pod seznamem dotazů zobrazí přehled nalezených dokumentů (Obrázek A.13). Tento seznam obsahuje několik informací o každém nalezeném dokumentu:

- Název.
- Celkové pořadí a skóre, které dokument dosáhl. Pokud jeden a ten samý dokument byl nalezen více servery, tyto hodnoty představují průměrnou hodnotu. Hodnoty, které dokument dosáhl na jednotlivých serverech jsou uvedeny ve sloupečku Servery. Zde je nutné poznamenat, že ne každý server vrací ohodnocení nalezených dokumentů.
- Lokace dokumentu může představovat například URL adresu nebo jiný identifikátor dokumentu (například u Oraclu je to jméno tabulky a hodnota primárního klíče dokumentu).
- Seznam zkratk serverů, které našly dokument s uvedením pořadí a skóre, jež dokument na těchto serverech dosáhl.

- Datумы vytvoření a modifikace dokumentu (pokud se tyto informace podařilo zjistit).
- Poznámka.
- Ukázka z dokumentu se zobrazuje pod výše uvedenými informacemi. Zobrazení této ukázky je možno vypnout pomocí volby Zobrazit/Ukázka dokumentu.

Se seznamem nalezených dokumentů je možno dále pracovat (Oddíl A.4.6). Jednotlivé dokumenty lze prohlížet (položka Otevřít dokument v kontextové nabídce seznamu dokumentů nebo stačí dvakrát kliknout na záznam) pomocí asociovaného prohlížeče. Další možností je uložení dokumentu na disk (položka Uložit dokument).

Spodní část okna obsahuje i další informace o průběhu vyhledávání, které jsou schovány pod dalšími záložkami:

Popis chyby	Typ	Server	Parametr	Poznámka
Chyba při získávání výsledků:ORA-06503...	Chyba při vyhledávání	Oracle (Mater)		
Parametr pro slovo v zázci nemá zadán v...	Chyba při vyhledávání	Google	Slovo v zázci(1)...	Parametr nebude ...

Obrázek A.14: Chyby vzniklé při vyhledávání

Server	Čas vyhledání	Všechno dok.	Nalezeno dok.	Chyby	Dotaz	Poznámka
Altavista	1.913 s	10	20	0	http://www.altavista.c...	squash & praha & (stud...
Centura	2.183 s	10	487	0	http://5.alted.centura...	"squash" AND "praha"...
Google	0.811 s	10	230	0	http://www.google.co...	filetype:pdf +squash +...

Název	Hodnota
Čas vyhledání	8.312 s
Dokumenty	20
Chyby	0

Obrázek A.15: Statistika vyhledávání

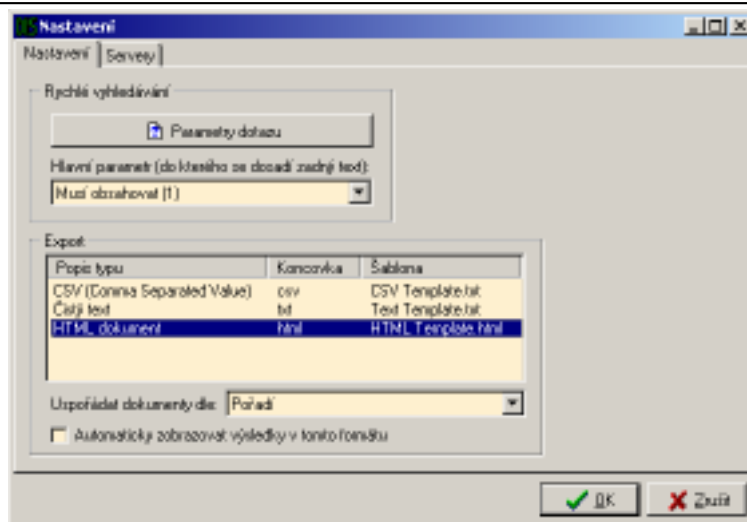
**Chyby** Seznam chyb (Obrázek A.14), ke kterým došlo v průběhu vyhledávání, případně chyby, které nebyly opraveny při editaci dotazu.

**Statistika** Obsahuje informace o celkové době hledání, počtu nalezených dokumentů, počtu chyb (Obrázek A.15). A také o tom, jak úspěšné byly jednotlivé servery:

- Jak dlouho jim hledání trvalo.
- Počet vrácených dokumentů.

- Kolik dokumentů celkem odpovídalo položenému dotazu (z těchto bylo pouze několik prvních vráceno).
- Počet chyb.
- Dotaz položený serveru. Tento řetězec obsahuje dotaz jak jej server dostal. Pro každý server je tento dotaz odlišný. Jedná-li se například o webový full-textový vyhledávač, je dotazem URL vyhledávacího stroje, která je doplněna příslušnými parametry. V takovém případě se po dvojitém kliknutí otevře asociovaný Internetový prohlížeč a zobrazí HTML stránku obsahující nalezené dokumenty, kterou oslovený server vrátil. Naopak například Oracle server vyplňuje toto políčko SQL dotazem.
- Poznámka obsahuje dodatek k vlastnímu dotazu a další doplňující informace.

### A.4.5 Nastavení

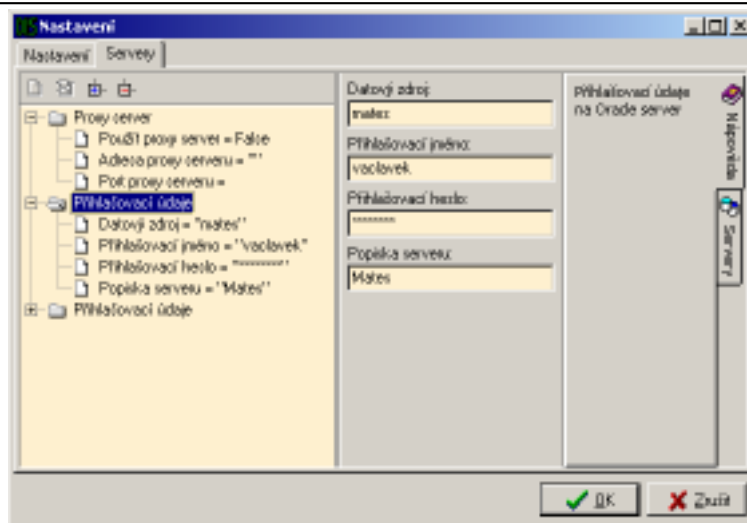


Obrázek A.16: Nastavení

Dialog pro nastavení (Obrázek A.16) umožňuje nastavit základní vlastnosti aplikace. Jedná se zejména o parametry rychlého hledání, zobrazování seznamu dokumentů a nastavení serverů.

Tlačítko Parametry dotazu na první záložce slouží k nastavení šablony dotazu pro rychlé vyhledávání. Umožňuje nastavit veškeré parametry dotazu. Hledaný řetězec (to co uživatel zadá) se dosadí do parametru, který je vybrán v comboboxu pod tímto tlačítkem.

Dále zde lze ještě určit jak se má zobrazovat uživateli seznam nalezených dokumentů při rychlém exportu. Na výběr je stejný seznam formátů jako při exportu (Oddíl A.4.6). Navíc je možno zvolit automatické zobrazování, které zapříčiní okamžité zobrazení nalezených dokumentů po vyhledání ve zvoleném formátu za použití asociovaného prohlížeče. To využijí uživatelé zejména v tom případě, že se jim nelíbí stávající zobrazování. Vzhled výsledného dokumentu lze změnit pozměněním stávající šablony nebo vytvořením nové (viz Oddíl A.5).



Obrázek A.17: Nastavení serverů

Druhá záložka (Obrázek A.17) umožňuje nastavit veškeré parametry všech dostupných serverů. S tímto dialogem se pracuje obdobně jako při specifikaci parametrů dotazu.

V současnosti lze nastavit parametry proxy serveru pro webové vyhledávače a přihlašovací údaje pro Oracle server. Poslední položka v této skupině parametrů (Popiska serveru) je uživatelský název serveru. Tento název bude ve všech seznamech identifikovat server.

#### A.4.6 Seznamy dokumentů, export

Seznamy nalezených dokumentů lze samozřejmě uložit pro pozdější práci (položka Dokument/Uložit seznam dokumentů) a zpětně načíst (položka Dokument/Načíst seznam dokumentů). O tom, zda k dotazu existuje uložený seznam nalezených dokumentů, informuje uživatele malá ikona diskety ve druhém sloupci přehledu.

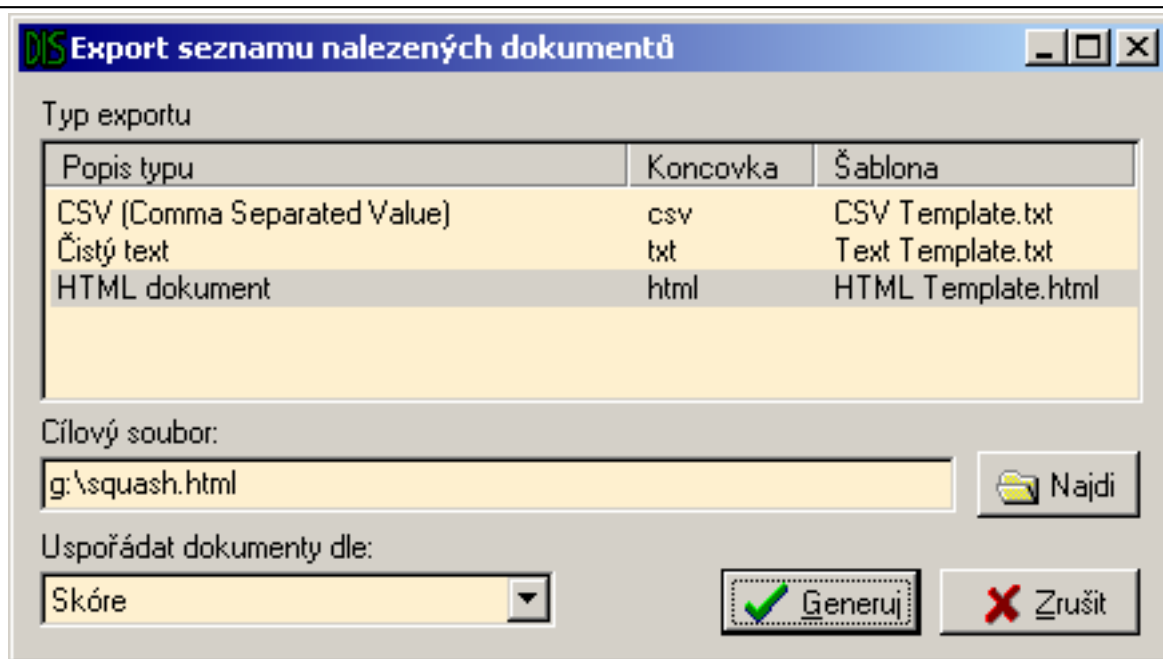
Jiným způsobem uložení seznamu dokumentů je jeho export (Dokument/Export seznamu dokumentů). Exportovat lze do několika formátů:

**HTML dokument (html)** Report vygenerovaný jako HTML stránka s použitím kaskádových stylů.

**Comma Separated Value (csv)** Tento formát lze načíst do většiny tabulkových programů jako je například Microsoft Excel (kde je možno s daty dále pracovat). Jednotlivé hodnoty jsou ve výstupním souboru odděleny čárkou, věty jsou odděleny odřádkováním.

**Čistý text (txt)** Univerzální formát, neobsahuje žádné formátovací tagy, platformě nezávislé.

Je zde také možnost přidat další šablony pro export. Stačí vytvořit příslušný textový soubor (popis viz Oddíl A.5) a uložit jej do podadresáře Export Templates. Na jeho jméno a koncovce nezáleží.



Obrázek A.18: Export seznamu dokumentů

Pro vlastní export (Obrázek A.18) stačí vybrat požadovaný formát, zadat jméno souboru (koncovka je nepovinná, pokud nebude uvedena, doplní se automaticky dle vybraného formátu) a pořadí v jakém mají být dokumenty na výstupu uspořádány. Po úspěšném vygenerování je možno výsledek ihned prohlédnout pomocí asociovaného prohlížeče.

Uživatel může také využít „rychlý export“, který slouží pouze pro prohlížení výsledků (výsledek se ukládá do pomocného adresáře). Není třeba opakovaně zadávat vlastnosti tohoto exportu (ty lze editovat v nastavení, viz Oddíl A.4.5).

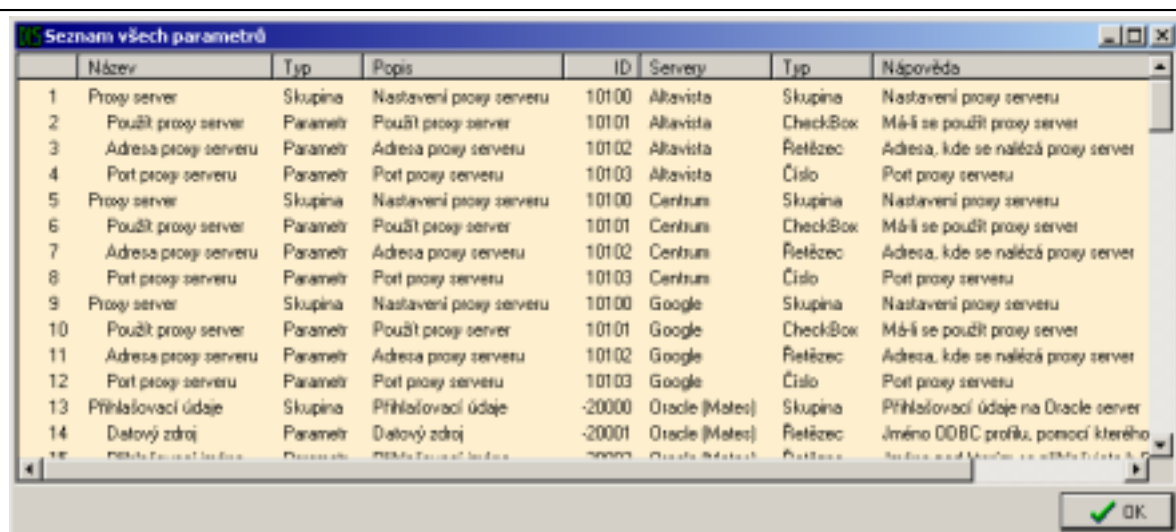
### A.4.7 Další možnosti aplikace

V aplikaci lze mírně měnit vzhled zobrazování. Veškeré volby týkající se této oblasti jsou v podnabídce Zobrazit. Lze vypnout zobrazování ukázky dokumentu (v seznamu dokumentů zabírá téměř vždy nejvíce místa), zobrazování seznamu dokumentů, případně seznamu dotazů.

Spíše pro vývojáře dalších DLL knihoven rozšiřujících spektrum vyhledávacích služeb je určen přehled (Obrázek A.19) všech implementovaných parametrů (Informace/Seznam parametrů).

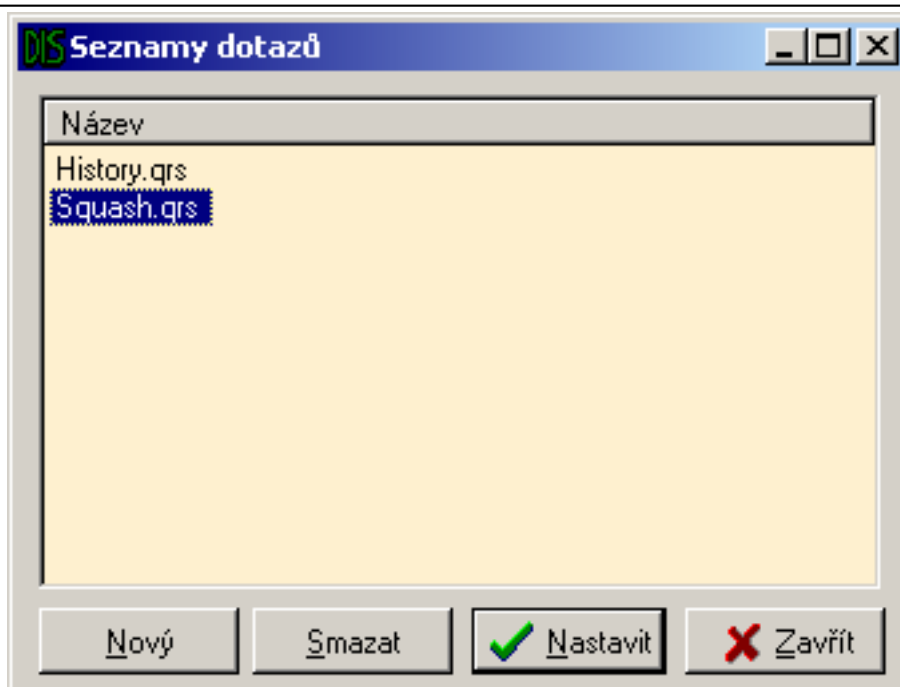
Pro přehlednější a rychlejší práci s aplikací je vhodné používat více seznamů dokumentů. Dotazy tak budou rozumně uspořádány a také se tím zrychlí start a ukončení aplikace, kdy dochází k načítání a ukládání historie dotazů.

Přehled seznamů dotazů (Obrázek A.20) se nachází pod položkou Dotaz/Seznamy dotazů. V tomto přehledu lze vytvořit nový, smazat stávající, či nastavit zvolený seznam jako aktuální.



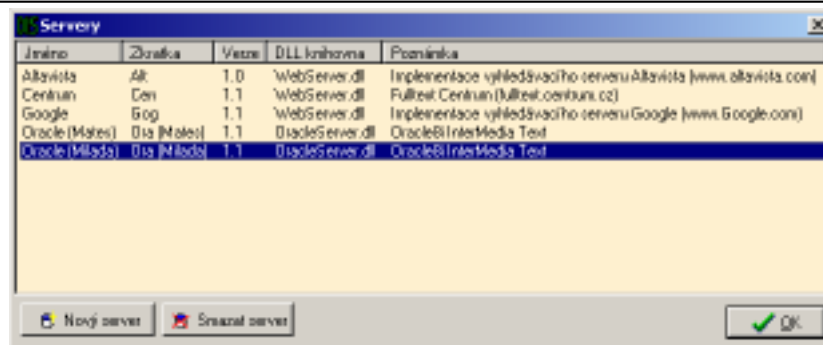
	Název	Typ	Popis	ID	Servery	Typ	Nápověda
1	Proxy server	Skupina	Nastavení proxy serveru	10100	Altavista	Skupina	Nastavení proxy serveru
2	Použít proxy server	Parametr	Použít proxy server	10101	Altavista	CheckBox	Má-li se použít proxy server
3	Adresa proxy serveru	Parametr	Adresa proxy serveru	10102	Altavista	Řetězec	Adresa, kde se nalézá proxy server
4	Port proxy serveru	Parametr	Port proxy serveru	10103	Altavista	Číslo	Port proxy serveru
5	Proxy server	Skupina	Nastavení proxy serveru	10100	Centrum	Skupina	Nastavení proxy serveru
6	Použít proxy server	Parametr	Použít proxy server	10101	Centrum	CheckBox	Má-li se použít proxy server
7	Adresa proxy serveru	Parametr	Adresa proxy serveru	10102	Centrum	Řetězec	Adresa, kde se nalézá proxy server
8	Port proxy serveru	Parametr	Port proxy serveru	10103	Centrum	Číslo	Port proxy serveru
9	Proxy server	Skupina	Nastavení proxy serveru	10100	Google	Skupina	Nastavení proxy serveru
10	Použít proxy server	Parametr	Použít proxy server	10101	Google	CheckBox	Má-li se použít proxy server
11	Adresa proxy serveru	Parametr	Adresa proxy serveru	10102	Google	Řetězec	Adresa, kde se nalézá proxy server
12	Port proxy serveru	Parametr	Port proxy serveru	10103	Google	Číslo	Port proxy serveru
13	Přihlašovací údaje	Skupina	Přihlašovací údaje	-20000	Oracle (Matec)	Skupina	Přihlašovací údaje na Oracle server
14	Datový zdroj	Parametr	Datový zdroj	-20001	Oracle (Matec)	Řetězec	Jméno ODBC profilu, pomocí kterého

Obrázek A.19: Přehled všech parametrů



Obrázek A.20: Seznamy dotazů

## A.4.8 Využívané servery



Obrázek A.21: Seznam dostupných serverů

Základními prvky celé aplikace jsou servery, které se využívají při vyhledávání. Jejich seznam (Obrázek A.21) je přístupný přes položku Nastavení/Dostupné servery. V seznamu jsou uvedeny nejen jejich názvy, ale i zkratky, které se zobrazují v některých stručnějším výpisech, DLL knihovna, ve které jsou implementovány a další informace.

Některé servery umožňují svou duplikaci a tím rozšiřují řadu použitelných zdrojů. Jedním z příkladů je server Oracle. Vytvořením jeho další kopie tlačítkem Nový server (je povolené v okamžiku, kdy je vybrán server umožňující vytváření svých kopií) vznikne další plnohodnotný zdroj pro vyhledávání, který se od originálu liší nastavením (to je možno upravit v dialogu Nastavení/Nastavení). Duplicitní servery lze samozřejmě také smazat (tlačítko Smazat server).

Jiným způsobem, jak rozšířit množinu použitelných serverů je získání další DLL knihovny a to buď od výrobce programu nebo od kohokoliv jiného. Nové knihovny je třeba nahrát do adresáře Servers, který je v hlavním adresáři aplikace.

## A.5 Vlastní šablona pro export

Výběr formátů na export je v dodávané verzi aplikace omezený (HTML, CSV, TXT). Uživatel si ale může kdykoliv vytvořit vlastní novou šablonu pro export do libovolných dalších formátů (XML, TEX, ...).

Tuto šablonu (textový soubor, jehož popis bude následovat dále) je třeba uložit do podadresáře Export Templates adresáře, kde se nachází vlastní exe soubor aplikace. Na jeho jméno a koncovce nezáleží.

### A.5.1 Popis souboru šablony

Šablonu exportu tvoří jednoduchý textový soubor. Kromě vlastního formátování výstupního exportu (například HTML tagy v případě HTML exportu) obsahuje i speciální tagy, které aplikace nahradí příslušnými daty. Tyto tagy mají následující syntaxi:

```
<#JMENO_TAGU JMENO_ATRIBUTU='HODNOTA' >.
```

Název tagu	Význam
ACT_DATE	Aktuální datum a čas (tedy datum a čas exportu)
USER_NAME	Jméno aktuálně přihlášeného uživatele
DOC_COUNT	Počet nalezených dokumentů
QNAME	Název dotazu
QCREATED	Datum a čas vytvoření dotazu
QEDITED	Datum a čas poslední modifikace dotazu
QSEARCHED	Datum a čas posledního vyhledávání dokumentů odpovídajících dotazu
QTOTAL_COUNT	Maximální počet nalezených dokumentů z jednoho serveru
QSERVER_COUNT	Počet dotazem využívaných serverů
QSERVERS	Seznam serverů, které dotaz používá pro vyhledávání
QNOTE	Poznámka k dotazu
DNUM	Aktuální číslo dokumentu
DNAME	Název dokumentu
DLOCATION	Lokace dokumentu
DSCORE	Celkové skóre dosažené dokumentem
DORDNUM	Celkové pořadí dokumentu
DCREATED	Datum vytvoření dokumentu
DMODIFIED	Datum poslední změny dokumentu
DEXTRACT	Ukázka z dokumentu
DNOTE	Poznámka k dokumentu
DSERVERINFOS	Informace o hodnocení dokumentu servery, které jej našly
TEMPLATE_INFO	Tag popisující vlastní šablonu, musí být uveden na začátku souboru (tedy jako první tag), má dva atributy.
LIST_BEGIN	Tag označující začátek smyčky, která obsahuje všechny nalezené dokumenty. Mezi tagy LIST_BEGIN a LIST_END by měly být uvedeny tagy popisující dokument, které budou nahrazeny skutečnými vlastnostmi nalezených dokumentů.
LIST_END	Konec smyčky dokumentů

Tabulka A.1: Názvy a popisy tagů šablony



Rozšiřující atributy jsou použity pouze u jediného tagu `TEMPLATE_INFO`, který popisuje šablonu a musí být uveden na samém začátku souboru. Popis atributů tohoto tagu:

**Ext** Koncovka, s jakou se bude soubor ukládat (`html`, `txt`, `xml`, ...). Tato koncovka je poměrně důležitá, neboť to bude výchozí koncovka, pokud uživatel nezapiše jinou a jednak se na zobrazení výsledného exportu použije prohlížeč, který je s touto koncovkou asociován.

**Name** Jméno (může být i s krátkým komentářem) pod jakým bude šablona uvedena v seznamu dostupných formátů pro export.

## A.6 Klávesové zkratky

Klávesová zkratka	Popis
<b>Ctrl+N</b>	Nový dotaz
<b>Ctrl+D</b>	Odvodit nový dotaz
<b>Ctrl+E</b>	Editace dotazu
<b>Shift+Del</b>	Smazání dotazu
<b>Ctrl+F</b>	Vyhledat
<b>Ctrl+S</b>	Uložit seznam dokumentů
<b>Ctrl+O</b>	Načíst seznam dokumentů
<b>Ctrl+T</b>	Zobrazit seznam šablonou
<b>Ctrl+F1</b>	Informace o aplikaci
<b>F1</b>	Nápověda
<b>Ins</b>	Přidání parametru (ve stromě parametrů)

Tabulka A.2: Klávesové zkratky

# Příloha B

## Obsah příloženého CD

`Index.htm` HTML stránky popisující obsah tohoto CD.

`DISClient.msi` Instalační soubor aplikace DISClient.

[`DISClient`] Adresář obsahující vlastní aplikaci, která se nemusí instalovat (stačí obsah tohoto adresáře překopírovat na cílové místo na disku) a spustit souborem `DISClient.exe`.

[`Documents`] Dokumentace k programu a vlastní diplomová práce v několika formátech (`.rtf`, `.pdf`, `.html`).

[`Servis`] Několik nástrojů, které by mohl uživatel potřebovat (Oracle8i klient, Adobe Acrobat Reader 5.0, Windows Installer Engine pro starší operační systémy).

[`Source`] Zdrojové texty programu a celé diplomové práce.